

AD A100135

**LEVEL**

DOCUMENTATION  
CAA-D-81-2

AD

A107016

**WARTIME REQUIREMENTS FOR  
AMMUNITION, MATERIEL, AND PERSONNEL  
(WARRAMP)**

**VOLUME II**

**MATERIEL POST PROCESSOR  
PROGRAM MAINTENANCE MANUAL  
(MPP-PMM)**



**DTIC**  
DEC 3 1 1981

**AUGUST 1981**

**PREPARED  
FOR**

**U.S. ARMY CONCEPTS ANALYSIS AGENCY  
8120 WOODMONT AVENUE  
BETHESDA, MARYLAND 20814  
BY**

**CACI, INC. - FEDERAL  
1815 NORTH FORT MYER DRIVE  
ARLINGTON, VIRGINIA 22209**

**CONTRACT NUMBER DA-003-81-0-0001**

**DISTRIBUTION STATEMENT A**  
Approved for public release  
Distribution unlimited

DTIC 81 021

The findings of this report are not to be construed as an official Department of the Army position, policy, or decision unless so designated by other official documentation. Comments or suggestions should be addressed to:

Commander  
US Army Concepts Analysis Agency  
ATTN: CSCA-MCM  
8120 Woodmont Avenue  
Bethesda, Maryland 20014

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER CAA-D-81-2 - <i>Volume 1</i>	2. GOVT ACCESSION NO. <i>AD-A109 135</i>	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Wartime Requirements for Ammunition Materiel and Personnel (WARRAMP), Volume II, Materiel Postprocessor Program Maintenance Manual		5. TYPE OF REPORT & PERIOD COVERED Final: Sep 80 - Aug 81
7. AUTHOR(s) Cpt. Scott Cantlon (USACAA) Mr. Ronald G. Rhoades (CACI, Inc.)		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS CACI, Inc. 1815 North Fort Myer Drive Arlington, VA 22209		8. CONTRACT OR GRANT NUMBER(s) MDA903-80-D-0668
11. CONTROLLING OFFICE NAME AND ADDRESS US Army Concepts Analysis Agency 8120 Woodmont Avenue Bethesda, Maryland 20014		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE August 1981
		13. NUMBER OF PAGES 447
		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report)  APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES ACCOMPANIED BY: Wartime Requirements for Ammunition, Materiel and Personnel (WARRAMP), Volume I: Materiel Postprocessor User's Manual		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)		
UTILITY PROGRAMS FORTRAN ASCII FORTRAN SIMSCRIPT II.5 SYMWAR	EQUIPMENT CLASSES WARF CONCEPTS EVALUATION MODEL COSAGE ELCON	LEA TAPES LINCODE WARRAMP WIMP LOSS RATES
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This manual provides information on the programs, processing and maintenance of the 19 computer software programs that comprise the Materiel Postprocessor (MPP) of the Wartime Requirements for Ammunition, Materiel and Personnel (WARRAMP) methodology. This manual provides a general overview of the methodology system followed by a programmer level discussion of each program. The discussion includes details on the description of processing, the operating environment and program maintenance procedures. Sample runstreams and program source code listing are included in this manual.		

DD FORM 1 JAN 72 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

DTIC  
ELECTED  
DEC 31 1981  
H

44-1135

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

20. ABSTRACT

for the applications on the UNIVAC 1100/82 installed at the US Army Concepts Analysis Agency.

Accession For	
NTIS GRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability	
Dist	Avail and Special
A	

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

**WARTIME REQUIREMENTS FOR AMMUNITION, MATERIEL, AND PERSONNEL  
(WARRAMP)**

**Volume II**

**MATERIEL POSTPROCESSOR  
PROGRAM MAINTENANCE MANUAL**

**(MPP-PMM)**

**September 1981**

**PREPARED  
FOR**

**U. S. ARMY CONCEPTS ANALYSIS AGENCY  
8120 WOODMONT AVENUE  
BETHESDA, MARYLAND 20014**

**BY**

**CACI, INC.  
1815 NORTH FORT MYER DRIVE  
ARLINGTON, VIRGINIA 22209**

**CONTRACT MDA903-80-C-0668**

AD-A109135  
WARTIME REQUIREMENTS FOR AMMUNITION, MATERIEL, AND PERSONNEL  
(WARRAMP)

Volume II

MATERIEL POSTPROCESSOR  
PROGRAM MAINTENANCE MANUAL

(MPP-PMM)

September 1981

PREPARED  
FOR

U. S. ARMY CONCEPTS ANALYSIS AGENCY  
8120 WOODMONT AVENUE  
BETHESDA, MARYLAND 20014

BY

CACI, INC.  
1815 NORTH FORT MYER DRIVE  
ARLINGTON, VIRGINIA 22209

CONTRACT MDA903-80-C-0668

**MATERIEL POSTPROCESSOR  
PROGRAM MAINTENANCE MANUAL**

**(MPP-PMM)**

**CONTENTS**

SECTION	PAGE
I      GENERAL .....	1
Purpose .....	1
Standards .....	1
Project References .....	2
Terms and Abbreviations .....	2
II     SYSTEM DESCRIPTION .....	5
General Description .....	5
Security and Privacy .....	5
System Application .....	5
III    PROCESSOR COMPONENTS - DETAILED DESCRIPTION ...	17
1    WARF GENERATOR MODEL - EQUIPMENT LOSS CONSOLIDATOR (ELCON) .....	17
Description of Processing .....	17
Operating Environment .....	24
Maintenance Procedures .....	24
2    WARF INTERMEDIATE MATERIEL PROCESSOR .....	49
Description of Processing .....	49
Operating Environment .....	82
Maintenance Procedures .....	82

3	WIMP/TOE-IN .....	169
	Description of Processing .....	169
	Operating Environment .....	172
	Maintenance Procedures .....	173
4	WIMP/LOSS RATES .....	181
	Description of Processing .....	181
	Operating Environment .....	184
	Maintenance Procedures .....	185
5	UTILITY: LEA/TAPE.....	195
	Description of Processing .....	195
	Operating Environment .....	196
	Maintenance Procedures .....	196
6	UTILITY: ITMID/TEMP.....	203
	Description of Processing .....	203
	Operating Environment .....	204
	Maintenance Procedures .....	204
7	UTILITY: ITMID/REC-A.....	213
	Description of Processing .....	213
	Operating Environment .....	215
	Maintenance Procedures .....	215
8	UTILITY: TOE/IST-RUN .....	227
	Description of Processing .....	227
	Operating Environment .....	228
	Maintenance Procedures .....	228

9	UTILITY: SCRUB/TOE .....	241
	Description of Processing .....	241
	Operating Environment .....	242
	Maintenance Procedures .....	243
10	UTILITY: TOE/ADD-PLTS .....	251
	Description of Processing .....	251
	Operating Environment .....	253
	Maintenance Procedures .....	253
11	UTILITY: RAM/MATRIX OR WIMP/MATRIX .....	263
	Description of Processing .....	263
	Operating Environment .....	265
	Maintenance Procedures .....	265
12	UTILITY: TOTAL/UNITS and WIMP/TOTAL-UNITS...	275
	Description of Processing .....	275
	Operating Environment .....	277
	Maintenance Procedures .....	277
13.	UTILITY: TOTAL/CATEGORY and WIMP/TOTAL-CAT .....	289
	Description of Processing .....	289
	Operating Environment .....	291
	Maintenance Procedures .....	291
14	UTILITY: SEARCH/ENGAGEREP .....	303
	Description of Processing .....	303
	Operating Environment .....	306
	Maintenance Procedures .....	306

15	UTILITY: CEM/DATA .....	323
	Description of Processing .....	323
	Operating Environment .....	329
	Maintenance Procedures .....	329
16	UTILITY: CEM/LOSSES .....	355
	Description of Processing .....	355
	Operating Environment .....	359
	Maintenance Procedures .....	359
17	UTILITY: COUNT/DIVISIONS .....	375
	Description of Processing .....	375
	Operating Environment .....	378
	Maintenance Procedures .....	378
18	UTILITY: CONTROL/COMPILER .....	391
	Description of Processing .....	391
	Operating Environment .....	395
	Maintenance Procedures .....	395
19	UTILITY: FINAL/REPORT .....	413
	Description of Processing .....	413
	Operating Environment .....	416
	Maintenance Procedures .....	416

APPENDIX	PAGE
A Project References .....	A-1
B Terms and Abbreviations .....	B-1
C Distribution .....	C-1

## FIGURES

FIGURE	PAGE
II.1.1	WARRAMP Overview .....6
II.3.1	MPP Flow - 1 .....7
II.3.2	MPP Flow - 2 .....8
II.3.3	MPP Flow - 3 .....9
II.3.4	MPP Flow - 4 .....10
II.3.5	MPP Flow - 5 .....11
II.3.6	MPP Flow - 6 .....12
II.3.7	MPP Flow - 7 .....13
II.3.8	MPP Flow - 8 .....14
II.3.9	MPP Flow - 9 .....15
III.1.1	ELCON I/O Structure .....27
III.1.2	ELCON Logic Flow .....28
III.1.3	ELCON Source Listing .....35
III.1.4	HOLD-XX Sample Data .....43
III.1.5	ITMID/FINAL Data Example .....44
III.1.6	CONTROL-XX Data Example .....45
III.1.7	RATES-XX/SEC-1 Data Sample .....46
III.1.8	RATES-XX/MONTHLY-WOL Data Example .....47
III.1.9	RATES-XX/MONTHLY-WL Data Example .....48
III.2.1	WIMP Processor I/O Structure .....54
III.2.2	Data Relationships - TYPE.UNIT .....55
III.2.3	Data Relationships _UNIT .....56
III.2.4	Data Relationships - HE.MUNITIONS .....57
III.2.5	Data Relationships - TYPE.BTRY .....58
III.2.6	Data Relationships - IC.MUNITIONS .....59
III.2.7	Data Relationships - SUBMUNITIONS .....60
III.2.8	CATTU Input Data Sample .....84

III.2.9	Equipment Input Data Sample .....	85
III.2.10	Unit Input Data Sample .....	86
III.2.11	PEM Input Data Sample .....	87
III.2.12	TBTRY Input Data Sample .....	88
III.2.13	BTRY Input Data Sample .....	89
III.2.14	MUNS Input Data Sample .....	90
III.2.15	SUBMUNS Input Data Sample .....	91
III.2.16	HELA Input Data Sample .....	92
III.2.17	WIMP/AT-1 Input Data Sample .....	93
III.2.18	WIMP/IN-AT Input Data Sample .....	94
III.2.19	WIMP Preamble Source Code Listing .....	95
III.2.20	WIMP Main Source Code Listing .....	109
III.2.21	Routine CAT.TU.INPUT Source Code Listing .....	110
III.2.22	Routine EQ.TE.INPUT Source Code Listing .....	112
III.2.23	Routine UNIT.INPUT Source Code Listing .....	113
III.2.24	Routine P.E.M.INPUT Source Code Listing .....	115
III.2.25	Routine TB.INPUT Source Code Listing .....	117
III.2.26	Routine BTRY.INPUT Source Code Listing .....	118
III.2.27	Routine MUNS.INPUT Source Code Listing .....	120
III.2.28	Routine SUBM.INPUT Source Code Listing .....	122
III.2.29	Routine HE.LA.INPUT Source Code Listing .....	123
III.2.30	Routine WARF.ARTY Source Code Listing .....	124
III.2.31	Routine ERROR.STOP Source Code Listing .....	127
III.2.32	WIMP Processor Program Structure .....	128
III.2.33	WIMPOUT Output Sample Data .....	129
III.2.34	Main Logic Flow .....	130
III.2.35	CAT.TU.INPUT Logic Flow .....	131
III.2.36	EQ.TE.INPUT Logic Flow .....	134
III.2.37	UNIT.INPUT Logic Flow .....	136
III.2.38	P.E.M. Input Logic Flow .....	143
III.2.39	TB.INPUT Logic Flow .....	146
III.2.40	BTRY.INPUT Logic Flow .....	149
III.2.41	MUNS.Input Logic Flow .....	152
III.2.42	SUBM.INPUT Logic Flow .....	156
III.2.43	HE.LA.INPUT Logic Flow .....	160
III.2.44	WARF.ARTY Logic Flow .....	162

III.2.4	ERROR.STOP Logic Flow .....	168
III.3.1	WIMP/TOE-IN I/O Structure .....	174
III.3.2	RAM/MATRIX Input Data Sample .....	175
III.3.3	UNITXXX Input Data Sample .....	176
III.3.4	WIMP/TOE-IN Output Data Sample .....	177
III.3.5	WIMP/TOE-IN Logic Flow .....	178
III.3.6	WIMP/TOE-IN Source Code Listing .....	179
III.4.1	WIMP/LOSS-RATES I/O Structure .....	186
III.4.2	WIMP/LOSS-RATES Runstream .....	187
III.4.3	WIMP/AT-X Input Data Sample .....	188
III.4.4	WIMP/TOTAL-CAT Input Data Sample .....	189
III.4.5	WIMP/LOSS-XX Output Data Sample .....	190
III.4.6	WIMP/RATES-XX Output Data Sample .....	191
III.4.7	WIMP/LOSS-RATES Logic Flow .....	192
III.4.8	WIMP/LOSS-RATES Source Code .....	193
III.5.1	LEA/TAPE Input/Output Structure .....	197
III.5.2	LEA/TAPE Program Logic Flow .....	198
III.5.3	LEA/TAPE Source Code Listing .....	199
III.5.4	LINCODE/LIST Input Data Sample .....	200
III.5.5	DENSITY/PROFILES Data Example .....	201
III.5.6	QUANTITY/PROFILES Data Example .....	202
III.6.1	ITMID/TEMP Input/Output Structure .....	205
III.6.2	ITMID/TEMP Program Logic Flow .....	206
III.6.3	ITMID/TEMP Source Code Listing .....	207
III.6.4	DENSITY/PROFILES Data Example .....	208
III.6.5	QUANTITY/PROFILES Data Example .....	209
III.6.6	LINCODE/LIST Data Example .....	210
III.6.7	ITMID/TEMP Output Data Example .....	211
III.7.1	ITMID/REC-A Input/Output Structure .....	216
III.7.2	ITMID/REC-A Program Logic Flow .....	217

III.7.3	ITMID/REC-A Source Code Listing .....	222
III.7.4	ITMID/TEMP Data Example .....	224
III.7.5	ITMID/FINAL Data Example .....	225
III.7.6	MISSED/ITMID - CODES Data Example .....	226
III.8.1	TOE/IST RUN Input/Output Structure .....	230
III.8.2	TOE/IST RUN Program Logic Flow .....	231
III.8.3	TOE/IST RUN Source Code Listing .....	233
III.8.4	FASTAL TOE MASTER Input Data Example.....	235
III.8.5	ARRAYED/UNITS Data Example .....	236
III.8.6	UNMATCHED/UNITS Data Example .....	237
III.8.7	TOE/IST RUN Data Output Example.....	238
III.9.1	SCRUB/TOE Input/Output Structure .....	244
III.9.2	SCRUB/TOE Program Logic Flow .....	245
III.9.3	SCRUB/TOE Source Code Listing.....	246
III.9.4	TOE/IST RUN Input Data Example .....	247
III.9.5	LINCODE/LIST Input Data Example .....	248
III.9.6	SCRUB/TOE Output Data Example .....	249
III.10.1	TOE/ADD-PLTS Input/Output Structure .....	255
III.10.2	TOE/ADD-PLTS Program Logic Flow .....	256
III.10.3	TOE/ADD-PLTS Source Code Listing .....	258
III.10.4	SCRUBBED/TOE Data Example .....	260
III.10.5	PLATOONS Data Example.....	261
III.10.6	FINAL/TOE Output Data Example.....	262
III.11.1	RAM/MATRIX Input/Output Structure .....	266
III.11.2	RAM/MATRIX Program Logic Flow.....	267
III.11.3	RAM/MATRIX Source Code Listing.....	269
III.11.4	ARRAYED/UNITS Data Example.....	271
III.11.5	ITMID/FINAL Data Example.....	272
III.11.6	FINAL/TOE Data Example .....	273
III.11.7	RAM/MATRIX Data Example .....	274
III.12.1	TOTAL/UNITS Structure .....	278

III.12.2	TOTAL/UNITS Logic Flow .....	279
III.12.3	TOTAL/UNITS Source Listing .....	281
III.12.4	WIMP/TOTAL-UNITS .....	282
III.12.5	RAM/MATRIX Data Example .....	284
III.12.6	ARRAYED LIST OF UNITS Data Example .....	285
III.12.7	MISSED/UNITS-XX Record Layout .....	286
III.12.8	TOTAL/UNITS-XX Record Layout .....	287
III.12.9	WIMP/UNITS-XX Output Data Example .....	288
III.13.1	TOTAL/CATEGORY Input/Output Structure .....	292
III.13.2	TOTAL/Category Program Logic Flow .....	293
III.13.3	TOTAL/CATEGORY Source Code Listing .....	294
III.13.4	WIMP/TOTAL-CAT Source Code Listing .....	296
III.13.5	RAM/MATRIX Data Example .....	298
III.13.6	TOTAL/UNITS Data Example .....	299
III.13.7	TOTAL/CATEGORY Output Data Example .....	300
III.13.8	WIMP/TOTAL-CATEGORY Output Data Example .....	301
III.14.1	SEARCH/ENGAGEREP Input/Output Structure .....	308
III.14.2	SEARCH/ENGAGEREP Program Logic Flow .....	309
III.14.3	SEARCH ENGAGEREP Source Code Listing .....	314
III.14.4	SCENARIO/XX Output Data Example .....	317
III.15.1	CEM/DATA Input/Output Structure .....	330
III.15.2	CEM/DATA Program Logic Flow .....	331
III.15.3	CEM/DATA Source Code Listing .....	337
III.15.4	CEM/DATA REPORT (73AMP-XX) Data Example .....	346
III.16.1	CEM/LOSSES Input/Output Structure .....	360
III.16.2	CEM/LOSSES Program Logic Flow .....	361
III.16.3	CEM/LOSSES Source Code Listing .....	365
III.16.4	SUB ROUTINE PRINTX Source Code Listing .....	366
III.16.5	CEM/LOSS1 Input Data .....	367
III.16.6	CEM/LOSSES Output Data Example .....	373
III.17.1	COUNT/DIVISIONS Input/Output Structure .....	379

III.17.2	COUNT/DIVISIONS Program Logic Flow.....	380
III.17.3	COUNT/DIVISIONS Source Code Listing.....	385
III.17.4	COUNT/DIVISIONS Output Data Example .....	390
III.18.1	CONTROL/COMPILER Input/Output Structure.....	396
III.18.2	CONTROL/COMPILER Program Logic Flow .....	397
III.18.3	CONTROL/COMPILER Source Code Listing .....	403
III.18.4	CONTROL/TEMP Input Data Example .....	407
III.18.5	COUNT/DIVISIONS Input Data Example .....	408
III.18.6	SCENARIO/XX Input Data Example .....	409
III.18.7	REDARTY/DEGR Input Data Example .....	410
III.18.8	CEM/LOSSES Input Data Example .....	411
III.18.9	CONTROL/XX Output Data Example .....	412
III.19.1	FINAL/REPORT Input/Output Structure.....	417
III.19.2	FINAL/REPORT Program Logic Flow.....	418
III.19.3	FINAL/REPORT Source Code Listing .....	421
III.19.4	ITMID/FINAL Input Data Example.....	423
III.19.5	RATES-XX/SEC-1 Input Data Example.....	424
III.19.6	RATES-XX/MONTHLY-WOL Input Data Example.....	425
III.19.7	RATES-XX/MONTHLY-WL Input Data Example .....	426
III.19.8	FINAL/REPORT Output Example.....	427

## TABLES

I.2.1	Compiler Requirements .....	3
II.3.1	MPP Input/Output File Matrix .....	16
III.2.1	WIMP Input/Output File Matrix .....	53

**THIS PAGE IS INTENTIONALLY LEFT BLANK**

## Section I

### GENERAL

1. **PURPOSE:** The purpose of this Wartime Requirements for Ammunition, Materiel, and Personnel (WARRAMP) Materiel Postprocessor (MPP) Programmer's Maintenance Manual (PMM) is to document, describe and illustrate the WARRAMP MPP at a sufficient level of detail so that programmers charged with the system's maintenance can quickly understand the purpose and operation of the system, isolate problems or areas which require modification, make corrections and put the system back into operation. Documentation for each of the 19 utility programs which constitute the MPP includes, a description, purpose and organization of the utility, a flow chart, example program listings, a discussion of the input and outputs used by the utility with examples of each, a variable dictionary which identifies and describes each of the variables used in the utility, a discussion of the environment in which the utility operates and the maintenance procedures to be followed.

2. **STANDARDS:** As noted in the previous paragraph, the MPP consists of 19 individual computer programs. For the most part, FORTRAN IV is the standard programming language used, as can be seen from Table I.2.1. However, several utilities have used ASCII FORTRAN, two in FORTRAN II, and in one case SIMSCRIPT II.5 within CAA. These utility programs have evolved over several years with the evolution and growth of the AMMO-RATES methodology and the WARRAMP methodology.

Recompilation of the MPP programs will require the use of the different compilers in the computer system library as follows:

FORTRAN IV and V	@ FOR, options	SI, RO
ASCII FORTRAN	@ FTN, options	SI, RO
SIMSCRIPT II.5	@ SIM25, options	SI, RO

Refer to UNIVAC Publication (UP) 8244.1, the ASCII FORTRAN Programmers Reference Manual; or UP 4060, the FORTRAN V (and IV with extensions) Programmers Reference Manual; or the SIMSCRIPT II.5 User's Manual for the UNIVAC 1100 Series Computer Systems for the specific compiler options, re-compilation procedures, and compiler error diagnostics. In the notation above SI refers to the Source-code input file-name, element-name. RO refers to the Relocatable Object code program elements. After re-compilation, the object code must be collected into an executable program by employing the @MAP system processor.

In general these utilities produce one output file (element). This file is normally given the same title as the utility which produced it. For example, the RAM/MATRIX output file is produced by the RAM/MATRIX utility. For the most part variable names used within the utilities are given self-explanatory names. However, in all cases in this document a variable dictionary is provided for each utility. In many cases arrays or matrices are used to hold data read into the utility from input files. This feature centralizes the reading of data within the utility. Similarly, data written into output files can normally be found to initially

reside in arrays.

The data formats for the files are discussed in Volume I of this set; therefore they are not repeated in this volume. The programmer should refer to volume I for the information.

3. PROJECT REFERENCES: Project references can be found in Appendix A. This documentation effort was achieved through contractor support to USACAA, by CACI, Inc., under contract MDA903-80-D-0668. The Contracting Officers Technical Representative (COTR) was Mr. Hugh Jones, Models Group, Methodology and Computer Support Directorate, USACAA. This manual is one of a series to document the WARRAMP Methodology's computer software. Volume I of the series contains the user's portion of the instructions on this software.

4. TERMS and ABBREVIATIONS: Terms and abbreviations are used throughout to facilitate communications of sets of words (acronyms) and analytical expressions common to the methodology and military operations research. A complete listing may be found in Appendix B of this manual. In addition, the full statement of the expression followed by the acronym or term in closed parenthesis is used throughout the manual on the first occurrence of its use.

# COMPILER REQUIREMENTS

## Programming Language

### -----FORTRAN-----

UTILITY	IV	V	ASCII*	SIMSCRIPT II.5**
ELCON	X			X
WIMP				
LEA/TAPE	X			
ITMID/TEMP	X			
ITMID/REC-A	X			
TOE/ISTRUN			X	
SCRUB/TOE			X	
TOE/ADD-PLTS			X	
RAM/MATRIX			X	
TOTAL/UNITS	X			
TOTAL CATEGORY	X			
SEARCH/ENGAGEREP	X			
CEM/DATA	X			
CEM/LOSSES	X			
COUNT/DIVISIONS	X			
CONTROL/COMPILER	X			
FINAL/REPORT	X			
WIMP/TOE-IN		X		
WIMP/LOSS-RATES		X		

\* ASCII FORTRAN LEVEL 9 COMPILER INSTALLED CAA, 2 APRIL 81

\*\* SIMSCRIPT II.5 VERSION 7.0 INSTALLED JULY, 1981

Table I.2.1

**THIS PAGE IS INTENTIONALLY LEFT BLANK**

## Section II

### SYSTEM DESCRIPTION

1. **GENERAL DESCRIPTION:** The WARRAMP Materiel Postprocessor is the materiel portion of the WARRAMP analytical methodology. It exists to merge and organize combat equipment loss data from historical sources and the combat simulation modeling sources. Once merged, computations are performed to produce the quantity of losses of combat materiel and an expression of, or rate of, loss of the equipment items over a hypothesized period of combat (warfare) time. The produced losses and respective rate of loss support the US Armys operations and planning functions and the budgeting process. As such, the programs documented herein are unique and have the sole application of supporting the WARRAMP methodology. The relationship of the MPP function to WARRAMP are depicted in Figure II.1.1. The MPP major components are highlighted with a heavy border.

2. **SECURITY AND PRIVACY:** The individual software components (programs) are cataloged as indicated under the detailed descriptions for each program. In each case, they are cataloged in the public mode for user access. User's are asked not to modify or edit (write) in the program files. In event alteration is required for a specific purpose, a potential user should copy the program to a file under his/her user identification, and then edit the file as desired. In event of error detection during use, the user is requested to note the error by program line and forward the proposed correction to the program custodian, so that the record program may be updated. Test (sample) data, either input or output and the programs contained herein are unclassified. Users must apply the appropriate security classifications to their data files and are responsible for the safeguard of printed matter accordingly. Users and programmer personnel are directed to the installation computer user's guide for the appropriate classification levels. Throughout the manual, reference is made to classified data files. The reference is made to the file content, not the classified file qualifier.

The evolution of the WARRAMP methodology has necessitated changes in the MPP software. This documentation supports and facilitates such change. It is incumbent on the program custodian to verify and validate changes made to the record versions of the programs and disseminate changes to this manual.

3. **SYSTEM APPLICATION:** Sections I and II of the MPP User's Manual (Volume I of the WARRAMP documentation set) provides a comprehensive description of the Materiel Postprocessor methodology as it relates to WARRAMP. The relationships of the MPP utility programs are presented in figures II.3.1 through II.3.9. The MPP processes are interrelated as depicted in the figures; the output of one program becomes the input to another program. The MPP data flow is presented in Table II.3.1 and assists in understanding the MPP processes. The specific functions of the MPP utility programs are addressed in Section III under individual chapters by utility program.

# WARRAMP OVERVIEW

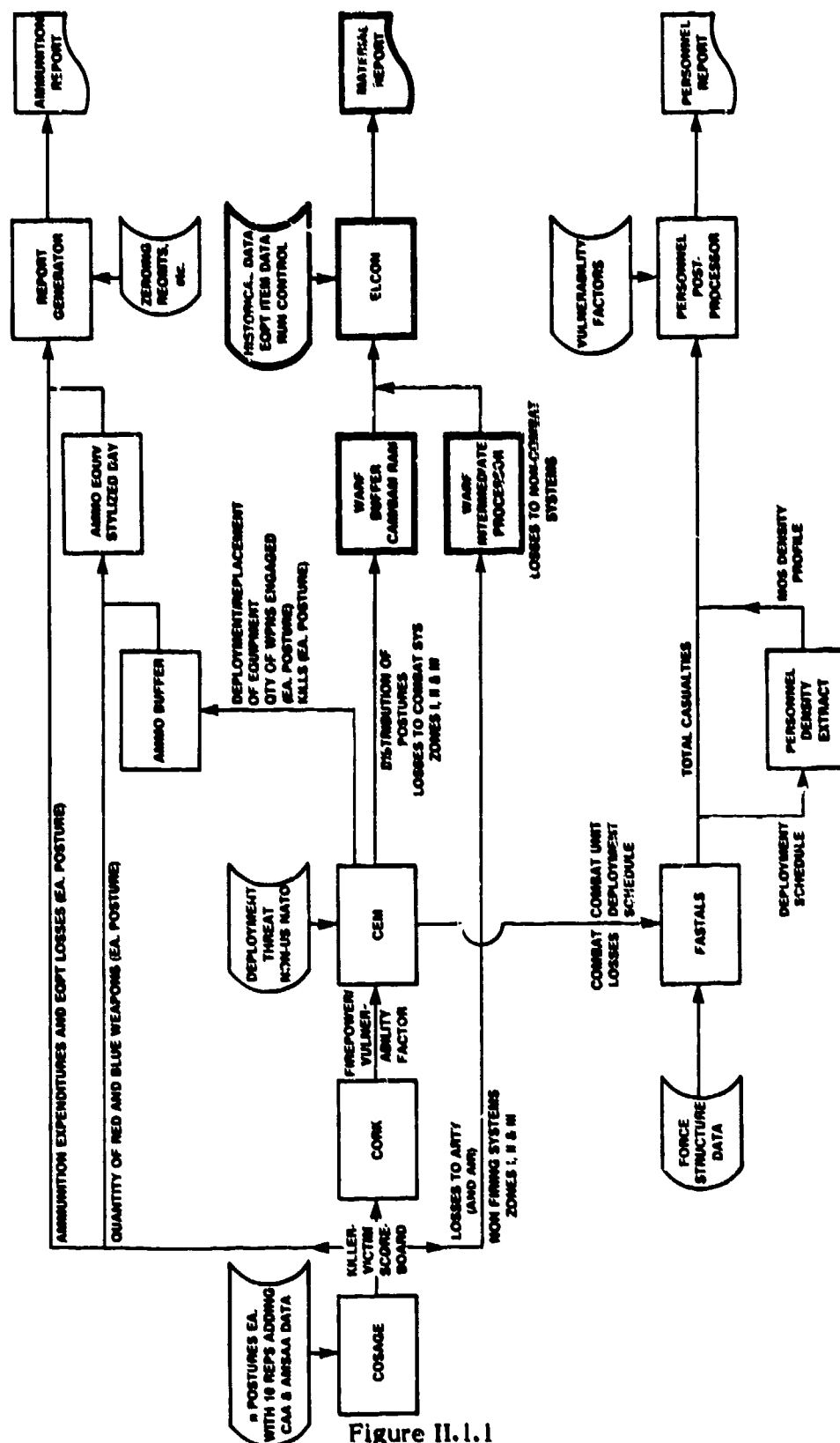


Figure II.1.1

MPP FLOW-1

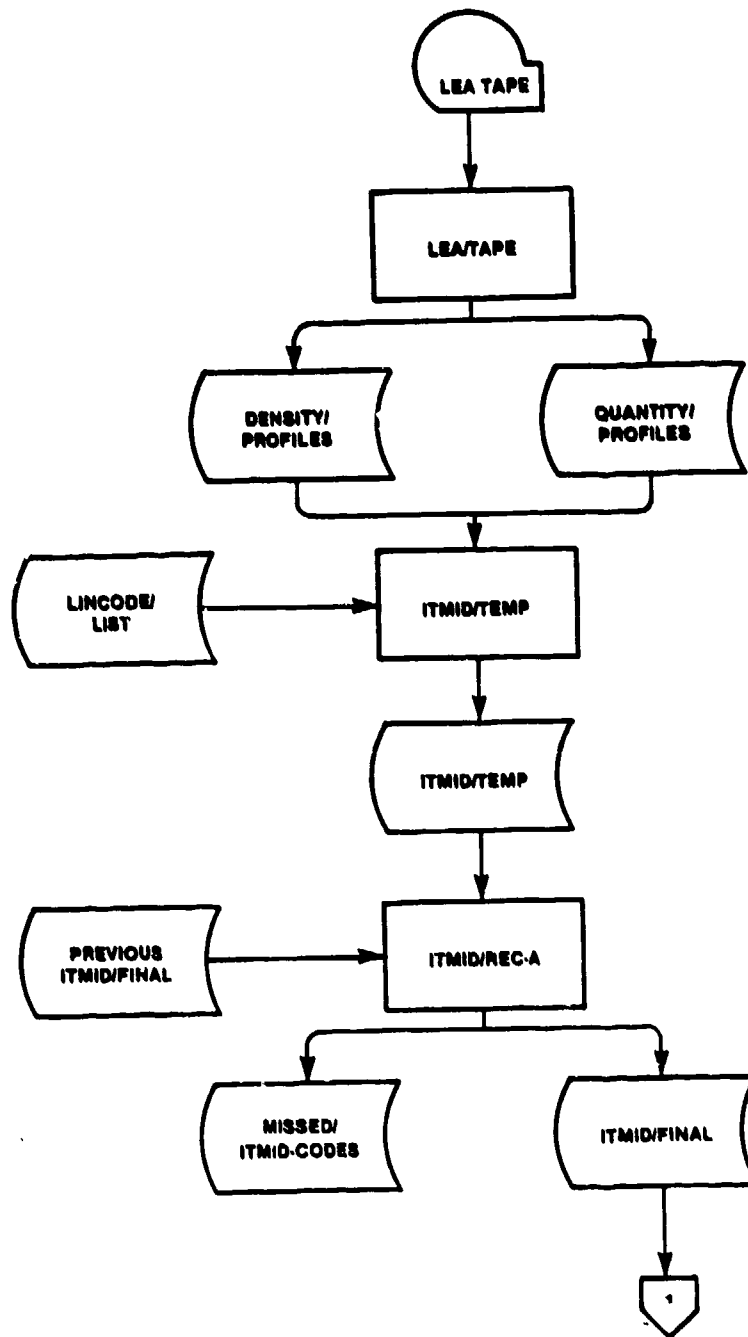


Figure II.3.1

MPP FLOW-2

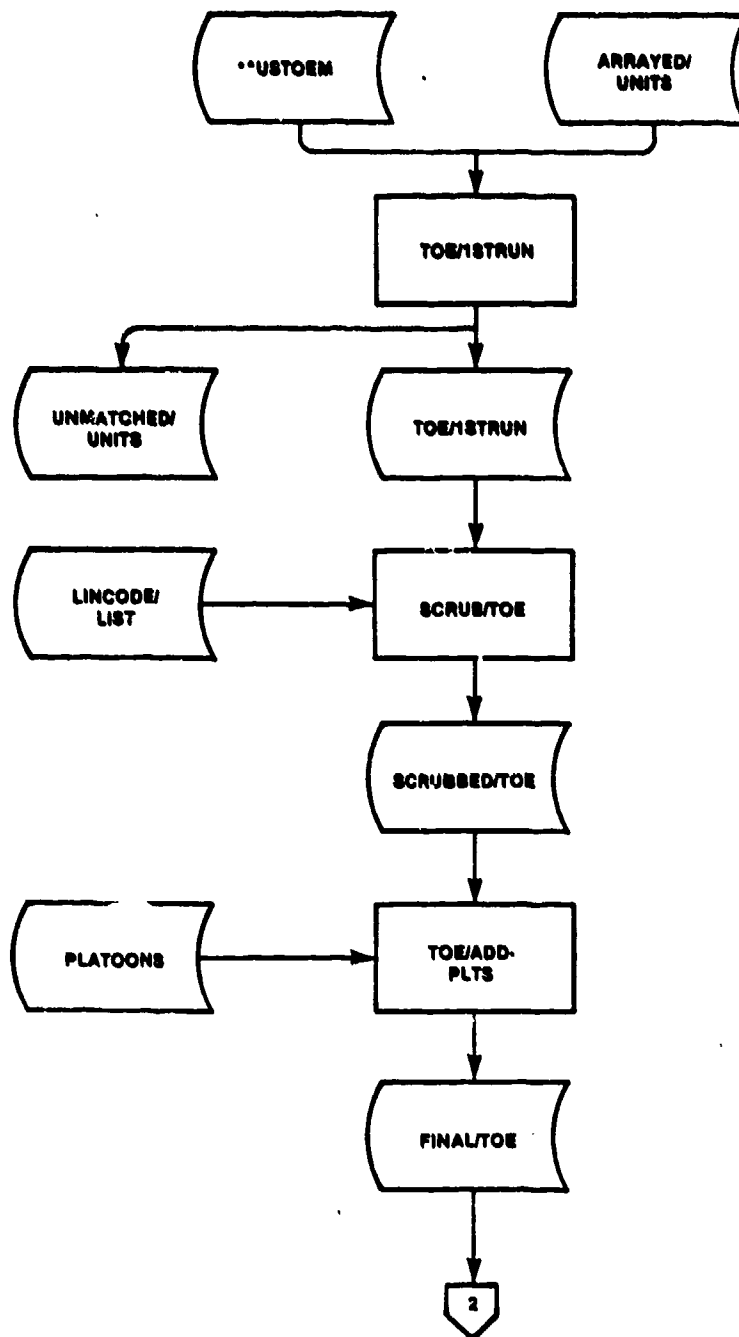
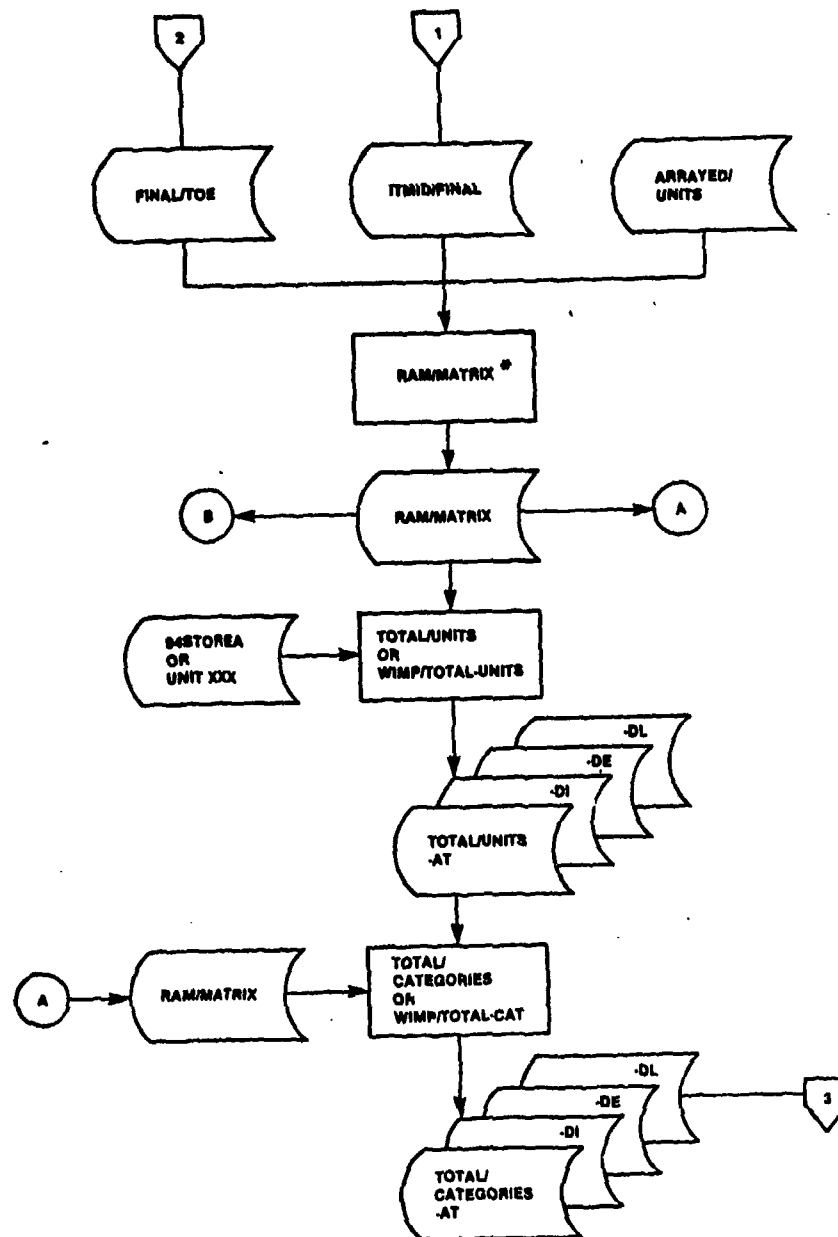


Figure II.3.2

MPP FLOW-3



\* The WIMP/MATRIX program may be employed in lieu.

Figure II.3.3

# MPP FLOW-4

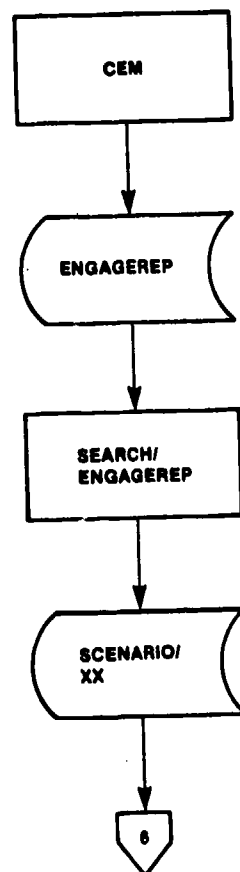
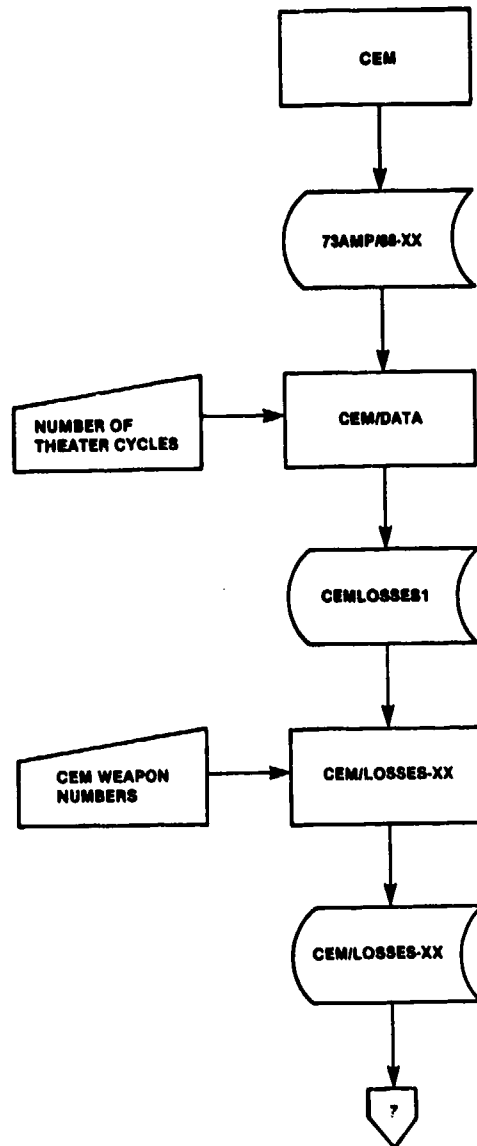


Figure II.3.4

**MPP FLOW-5**



**Figure II.3.5**

# MPP FLOW-8

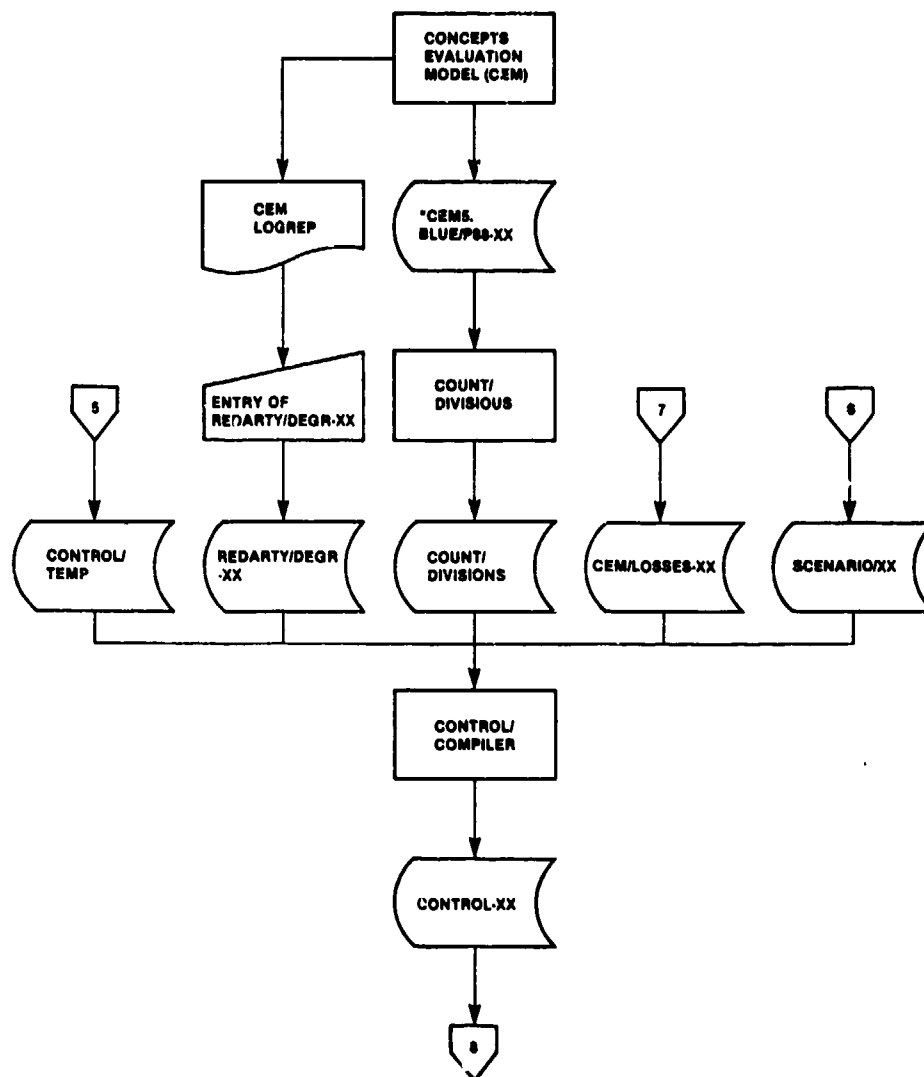


Figure II.3.6

# MPP FLOW-7

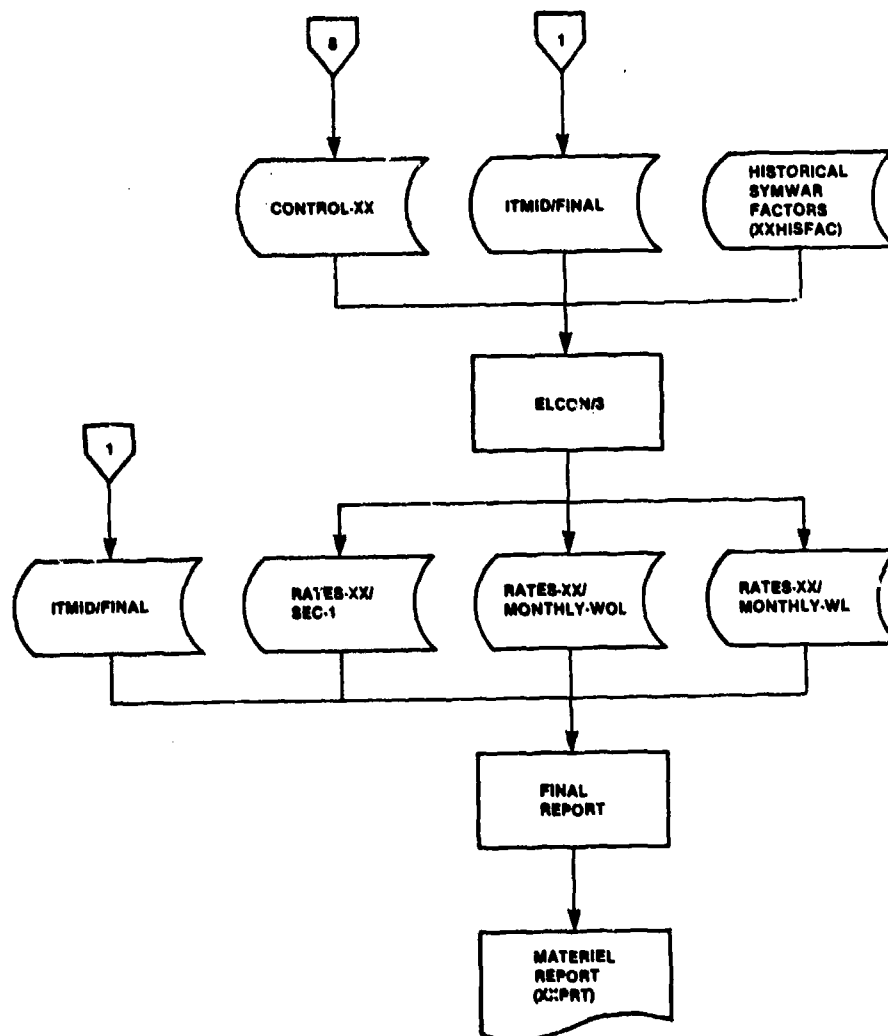


Figure II.3.7

# MPP FLOW-8

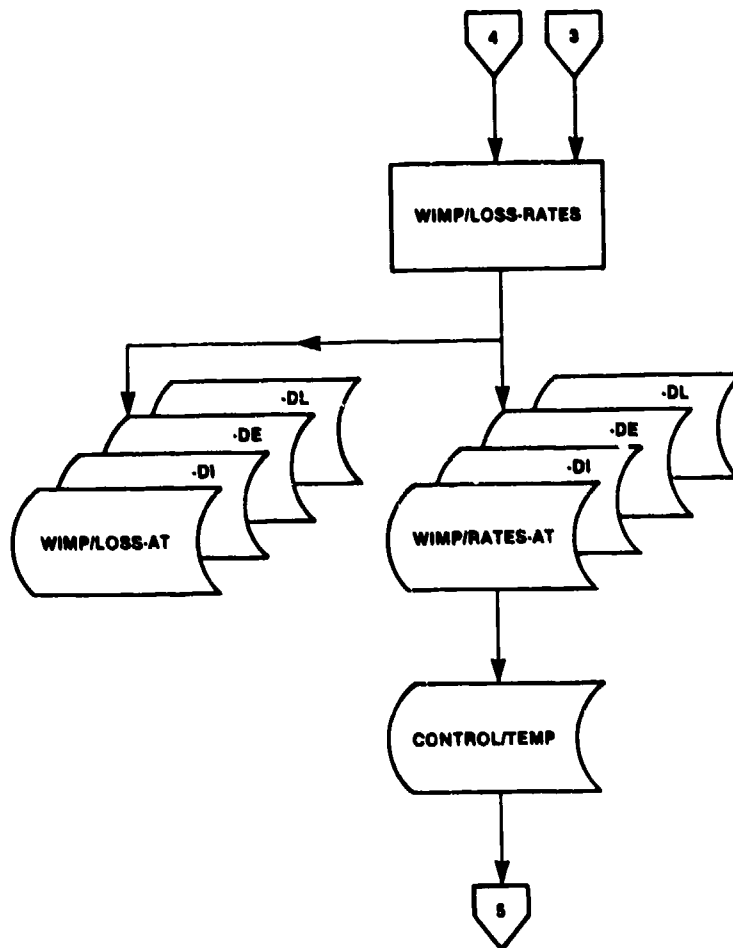
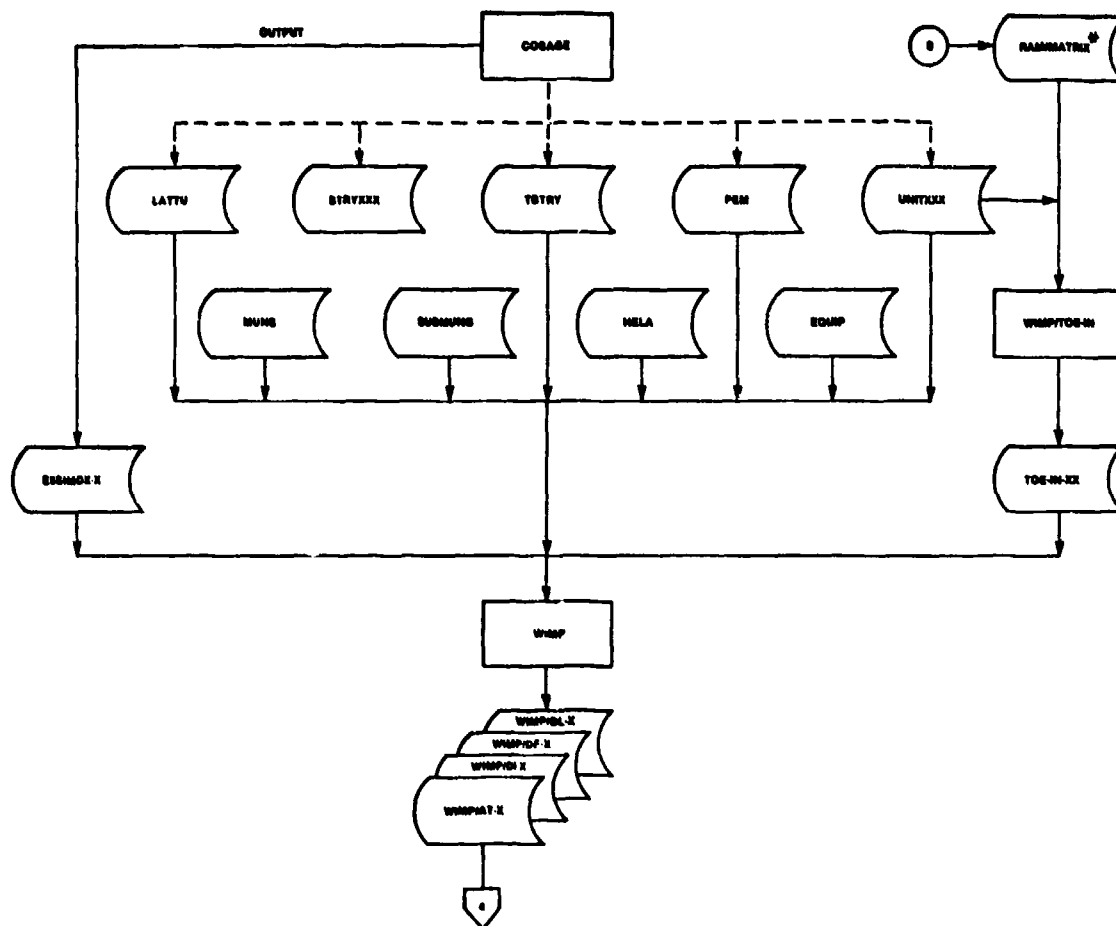


Figure II.3.8

MPP FLOW-8



\* The WIMP/MATRIX program may be employed in lieu.

Figure II.3.9

[illegible]

## SECTION III

### Chapter I

#### EQUIPMENT LOSS CONSOLIDATOR (ELCON)

1.1 DESCRIPTION OF PROCESSING: A complete discussion of the previously documented ELCON process can be found in superseded document CAA-D-79-3, August, 1979, entitled "Equipment Loss Consolidator (ELCON)" as referenced in Appendix A (referenced). The current documentation reference is item Y in Appendix A.

1.1.1 PURPOSE/FUNCTION: The purpose of the Equipment Loss Consolidator (ELCON) is to combine equipment loss data from historical and computer simulation sources with current equipment deployment and vulnerability information, as calculated in the MPP, in order to compute expected monthly loss rates for a list of specific equipment items.

1.1.2 PROGRAM INPUT/OUTPUT STRUCTURE: The overall structure of the utility is pictured in FIGURE III.1.1. The general logic flow within the utility is the flow chart, FIGURE III.1.2. An actual copy of the source code listing is contained in FIGURE III.1.3.

1.1.2.A INPUT and Data Base: The ELCON program requires three data input file elements. These files are the Historical SYMWAR Factors file, ITMID/FINAL file and the CONTROL/XX file. Each file is discussed below:

- o Historical SYMWAR Factors file - The equipment loss data found in this file is derived from analysis of WWII and Korean War experiences. This file is cataloged under the program file 82HISFAC. within the system. As the data here reflect historical experiences it is static, thus the user does not update it. A detailed explanation of the file structure, record layouts and data examples can be found in reference item Y, or the USACAA document CAA-D-79-3, ELCON, pages 4-1, 4-2, and C-1 through C-4.
- o ITMID/FINAL - This file contains data which describes each type of equipment in the battle area giving for example, its LINCODE, nomenclature, vulnerability category, etc. Also provided is the quantity of this item authorized for each of the seven time periods of the study and the density or distribution fraction of this item within each of the five zones of the combat area for each of the seven time periods. This file is a product of the ITMID/REC-A utility. Figures III.1.5 presents the data example of the file.
- o CONTROL-XX - This file contains such information as run control data, the equipment loss information provided by the CEM combat simulation (rather than historical experiences), Blue MIE losses to Red artillery, Blue Division strengths, etc. This file is a product of the CONTROL/-COMPILER utility. A complete discussion of this file can be found in

the CONTROL/COMPILER, Chapter 16 of this document. The data can be seen in Figure III.1.6.

**1.1.2.B OUTPUT Data and Data Files:** The ELCON program produces four output files. These files are the HOLD-XX, RATES-XX/SEC-1, RATES-XX/MONTHLY-WOL, and RATES-XX/MONTHLY-WL. The XX portion of the file name will be replaced by the appropriate CEM Run Control Number obtained from the CEM Operator/Analyst. The user will change the file names to reflect this CEM run control number in the runstream using the system editor. Each output file will be discussed below.

- o HOLD-XX - This is an interim file produced by the utility which contains the LOSS MATRIX for each period. (Figure III.1.4) The file is listed in machine readable code, not field data. The utility only creates this matrix if the variable WRITE F is equal to Zero. If WRITE F is not equal to 0 an existing matrix will be used by the utility. The WRITE F variable is read into the utility from the run parameter record (i.e. 2nd record) of the CONTROL/XX file. In creating and manipulating the matrix the utility uses features and commands outside of FORTRAN which are unique to UNIVAC EXEC 8 such as DEFINE, and FIND to construct the matrix the utility:
  - oo Calculates the maximum number of records that will be held in the matrix (line 179).  
$$KFSIZE = MAXITM * NPER * 4$$

Where:

KFSIZE = number of records  
MAX ITM = maximum number of equipment items being studied.  
NPER = number of time periods  
4 = number of combat postures.
  - oo Establish the file on Unit 2 (line 180). DEFINE FILE 2 as (KFSIZE,10, U, IVAR)  
Where:  
KFSIZE = number of records  
10 = record size in words. There are 10 causes that losses can be attributed to.  
U = The records will be unformatted  
IVAR = The value stored in this variable will contain the pointer to the current record.
  - oo Calculate the appropriate position within the file where the next record is to be written (Line 234).  
$$ILOC = (ISEQ-1) * NPER * 4 + (IPER-1) * 4 + 1$$

Where:

ILOC = The location within the file where the next loss record will be written.  
ISEQ = The sequence number assigned to each equipment item. This value is read in from the ITMID/FINAL file.  
NPER = Number Time periods in the study.  
4 = The number of combat postures.

IPER = The number of the current period being processed.

- oo After calculating the pointer ILOC the utility FINDS the record.
  - oo Calculates the entries for the record
  - oo Writes the record to the file
  - oo Sets ILOC equal to the returned pointer found in IVAR.
- o RATES-XX/SEC-1 - This file will be produced only if the PRINT INDICATOR, i.e., Columns 31-35 of the Run Parameters Record in the CONTROL-XX file is set to either "0" or "1". If it is set to "- 1", this file will not be produced. This file is used to collect output for the ELCON program and format it for printing, so that it is understandable by the combat analyst. As noted above, the user will have the option to specify the level of detail that will be contained in the file. If the user enters a "1" in the Print Indicator file of the Run Parameter Record the user will obtain a detailed report as output. This report will contain:
- oo A replay of the CONTROL-XX file that was used as input to the program with the addition of heading information explaining the data which follows it.
  - oo A detailed analysis of the loss rates consolidated from the historical data and the results of the theater simulation for each piece of major equipment being played in the study. Included here is the density profiles, loss rates from each of the 10 historical causes, losses in depot, etc. A complete example of a typical report can be found in the referenced ELCON documentation.

If the user enters a "0" into the Print Indicator field, a shortened version of the file will be prepared. This version will include the formatted replay of the CONTROL-XX file identical to that portion of the detailed output discussed above plus an item by item, one-line summary of each piece of equipment line code, name, WARF set (which is defined as a continuous period of time within the model delineated by a start period and an end period) and the loss rates broken out by in-theater losses and intertheater losses. An example of this output can also be found in the reference document, pages D-13 through D-15.

- o RATES-XX/MONTHLY-WOL - This file is always produced by the ELCON program. The file details for each item of equipment in the study the monthly in-theater loss rates (excluding LOC and depot logistic or shipping losses) for each item or time frame specified by the user. Figure III.1.8 presents example of the data found in the file.
- o RATES-XX/MONTHLY-WL - This file is similar to the above WOL file except that it contains the total loss rates for LOC, depot and inter-theater logistic (shipping) losses for each item identified by the user. This file always is produced by the ELCON utility. Figure III.1.9 depicts the example of the data contained in the file.

These three output files will be used in conjunction with the ITMID/FINAL file, produced by the ITMID/REC-A utility (Chapter 5) to produce the Final/Report which expresses loss rates in daily terms.

1.1.2.C DATA ELEMENT DICTIONARY: The following section identifies and defines the variables used in this utility.

<u>Name</u>	<u>Definition</u>
IHIS	This integer variable is used in READ statements to identify the proper unit from which the input file is read. In this case the variable is set to 4 and from the runstream unit 4 has been assigned to the **HISFAC file.
IN	This integer variable is used in READ statements to identify the second unit from which input data will be supplied. In is set to 5 and from the runstream unit 5 has been assigned the CONTROL/COMPILER and ITMID/FINAL files.
IO	An integer variable used in WRITE statements to identify the unit to which output is directed. IO is set to 6. From the runstream unit 6 is assigned to the output file RATES-XX/SEC-1. Do not confuse IO with 10.
ZONFAC	This is a 2 dimension array in the utility which holds the weighting factors used in loss computations. The first dimension identifies each of the five zones used in describing the combat area, the second dimension identifies each of the 10 loss causes. Entries in matrix are read from the * * HISFAC file.
CLRAT	This is a 3 dimensional matrix used by utility to organize and hold loss rates read from the * *HISFAC input file. The first dimension identifies each of the 10 causes of loss, the second dimension identifies each of the four combat postures; the third dimension each of the 36 historical classes used to categorize losses derived from historical sources.
REARL	This is a 2 dimensional matrix which is used to organize and hold the loss rates in two combat zones furthest from the FEBA (i.e., zones 4 and 5) or the REAR. The first dimension identifies each of the 36 historical classes; the second dimension identifies each of the 4 postures in which the unit is configured. Loss

rates in these last two combat zones are not currently generated from combat simulations. Thus loss data generated from World War II, Korea and Vietnam experiences are used to estimate these depot losses. These loss rates do not include losses from MINES, WEAROUT or ACCIDENT.

NPPRES	This run-parameter variable denotes the number of time periods for which replacement equipment is pre-stocked. This value is read from the input file CONTROL/COMPILER.
IWUL	This is an integer variable used to identify each of the 22 artillery vulnerability classes into which items of equipment are placed.
IPOS	This is an integer variable used to identify each of the four combat postures.
KFSIZE	This variable is calculated within the program in order to determine the size of the output file HOLD-XX. $KFSIZE = \text{Maximum number of items considered times the number of distinct time periods times the number of combat postures. (KFSIZE = MXITM * NPER * 4)}$ .
KATOEM	This is an integer variable read from the ITMID/FINAL file. This entry is a code assigned a specific item of equipment which categorizes as a specific type of equipment for losses from the CEM Theater Model.
ITEST	This is a temporary integer variable set to one less than the IPER value.
FRFWD	This variable is used as an array. Each occurrence of the array reflects for each time period in the study the fraction of the authorized equipment found in the forward three combat zones. This variable is calculated by the utility using the items DENS PRO array.
IWP	This is an integer variable established as an array within the utility. Entries in the array reflects the first time period of specific WARF sets. As is currently configured up to 20 WARF sets can be specified. This data is read in from the CONTROL/XX file on line 134.
NDPER	This is an integer variable which is used as an array to specify the number of days in each time period of the study. This data is read from the CONTROL/XX file on line 128.

KFIRST  
KLAST

These two integer variables are set to the current array valves found in IWP and LWP respectively and are used as parameters in following DO-LOOPS.

INOSHP

An integer variable used in WRITE statements to identify the unit to which output is directed. INOSHP is set to 7. From the runstream unit 7 is assigned to the output file RATES-XX/MONTHLY-WOL.

IWSHP

An integer variable used in WRITE statements to identify the unit to which output is directed. IWSHP is set to 8. From the runstream unit 8 is assigned to the output file RATES-XX/MONTHLY-WL.

TCAUS

A 12 character alphanumeric explanation of the cause of the loss. TCAUS is dimensioned as an array of 10 occurrences. The 10 loss causes are:

- 1 - DIRECT FIRE
- 2 - AREA FIRE
- 3 - MINES
- 4 - BOMBING
- 5 - STRAFFING
- 6 - ABANDONMENT
- 7 - WEAROUT
- 8 - ACCIDENT
- 9 - PILFERAGE
- 10 - GUERRILLA

DEPFAC

This is a variable used to accumulate losses of this item suffered during inter-theater transit of replacement equipment in the rear area. The value accumulated is the product of the losses in the rear area for this item's historical class and the current combat posture multiplied by the loss rate for a specific period and current combat posture suffered during inter-theater transit of replacement equipment. Unit of measure is items of equipment.

TOTAL

A variable used in the utility to accumulate the total losses suffered by this item. Unit of measure is equipments.

PCTDEP

This variable is calculated by the utility on line 405. The value of this variable reflects the monthly depot percentage loss rate for this item.

FRUNL

This variable is calculated by the utility on line 406 and reflects monthly unit loss rates for this item normalized to days of exposure. Unit of measure, fraction of losses per month, of this item.

PCTUNL This variable is simply FRUNL expressed as a percentage.

PCTRPL This variable is calculated on line 427 of the utility and reflects the loss percentage rate suffered by this item during inter theater transportation.

XXX A temporary variable which expresses for, in one instance (line 382), the ratio of the loss rate experienced during inter-theater transit of replacement parts for a specific period (RPLOS(IPER)) divided by the NON-loss Rate (i.e.  $1 - RLOS$ ). In another instance it represents ratio of air losses, (RAIR) divided by non-losses. It assumes the loss rate will never equal 1. XXX may take on values from 0 to 100.

YYY Same as XXX except it expresses the ratio for sea losses (RSEA) divided by non losses.

XLOC This is the loss rate for inter-theater transfer. This rate is calculated for each period during a WARF set for each item. For each period XLOC is calculated by:

Multiplying the item Density for this period by the number of days in the period. The resulting product will be the Days of exposure for this period for this item. This product is then multiplied by the XXX ratio. This is the ratio of the loss rate of replacement equipment on the inter-theater LOC for this period, divided by the NON-loss rate of this item in this period. This resulting product is then divided by the DAYS of Exposure for this item for the entire WARF set.

If the loss rates are relatively small the XXX ratio will not vary from the actual loss rate, however, as the rate grows larger, the XXX ratio will grow dramatically. Refer to the following table:

LOSS RATE	XXX	CHANGE
.001	.001001	+0.000001
.01	.010101	+0.000101
LOSS RATE	XXX	CHANGE
.1	.111111	+0.011111
.2	.25	+0.05
.3	.429	+0.129

.5	1.0	+0.5
.7	2.33	+1.63
.8	4.0	+3.2
.99	99.0	+98.01

1.2 OPERATING ENVIRONMENT: This program is implemented on the UNIVAC EXECUTIVE-8 operating system.

1.2.1 SUPPORT SOFTWARE: This routine requires the FORTRAN IV compiler and the system editor.

1.2.2 I/O DEVICES: The utility will receive its input from files resident on disk and in turn writes its output to disk.

1.3 MAINTENANCE PROCEDURES: The program is maintained on the system by the MPP analyst.

1.3.1 PROGRAMMING CONVENTIONS: Standard FORTRAN programming conventions are followed.

1.3.2 VERIFICATION PROCEDURES: Program verification is achieved through visual inspection of and the hand calculations of specific data gleaned from a sample program run.

1.3.3 ERROR CORRECTION PROCEDURES: The utility is structured to detect 4 error conditions during the reading of input data. These error conditions their causes and solutions are detailed below.

#### ERROR I

Message: SEQUENCE # (item seq.no.) LARGER THAN MAX SPECIFIED

Cause: The sequence number assigned to this equipment item is greater than the maximum specified by the user in the second field of the Run Parameter Record of the CONTROL-XX file.

Solution: Using the system editor access the ITMID/FINAL file and determine the sequence number of the last equipment item of the file. This number is denoted by the last field of the "A" or header record of the equipment item. With this number again using the system editor, access the Run Parameter Record of the CONTROL-XX file and change the maximum sequence number.

#### ERROR II

Message: SEQUENCE # (item seq. no.) MARKED AS CEM ITEM W/O CAT CEM

Cause: The header or "A" record of the item of equipment in the ITMID/FINAL file indicates that the combat losses for this item will come from CEM but the CEM category used to identify the type of equipment for losses from the theater model (CEM) has not been specified.

**Solution:** Notify CSSEM Operator/Analyst of discrepancy. If in fact this item is a CEM item, user must determine the appropriate CEM category and, using the system editor, enter in the appropriate CEM category type code for this item in the third field (i.e., positions 40-41, left justified) of the item's header record of the ITMID/FINAL file.

If it is determined that the item is not a CEM item the user must again access the header record of the item in the ITMID/FINAL file and change the last field (i.e., positions 53-54, left justified) to a 2 (if combat losses are from artillery) or a 3 (if combat losses are from historical data).

#### ERROR III

**Message:** SEQUENCE # (Item seq. no.) MARKED AS ARTY ITM W/O CATART

**Cause:** The header or "A" record of the item in the ITMID/FINAL file has indicated in the ninth field (i.e., positions 53-54) that the losses of this item will be from artillery but no artillery vulnerability class has been specified for the item in the fourth field of the record (i.e., positions 42-43).

**Solution:** Determine whether or not losses for this item are in fact to be determined by artillery. If not using the editor, change the ninth field (i.e., positions 53-54, left justified) to the appropriate loss code 1 = theater simulation (CEM); 3 = historical.

If the loss is to be from artillery, determine the appropriate vulnerability category for the item from the 22 available and using the editor enter it into the fourth field on the header record (i.e., positions 42-43, left justified).

#### ERROR IV

**Message:** ITEM # (item seq. no.) HAS ILLEGAL HISTORICAL CLASS

**Cause:** For the program to reach this error message the ninth field (positions 53-54) of the header record of the ITMID/FINAL file must be set to 3 (i.e., all losses are from history) and the historical class identifier is not between 1 and 36 and is thus illegal. This error, unlike the other 3, will not cause the program to stop execution.

**Solution:** Using the Item Sequence Number and the system editor examine the header record for this item, determine the appropriate historical class (i.e., a number from 1 to 36) and enter it into columns 53-54 right justifying the entry.

**1.3.4 Special Maintenance Procedures:** There are no established special or unique maintenance procedures for the program.

# ELCON STRUCTURE

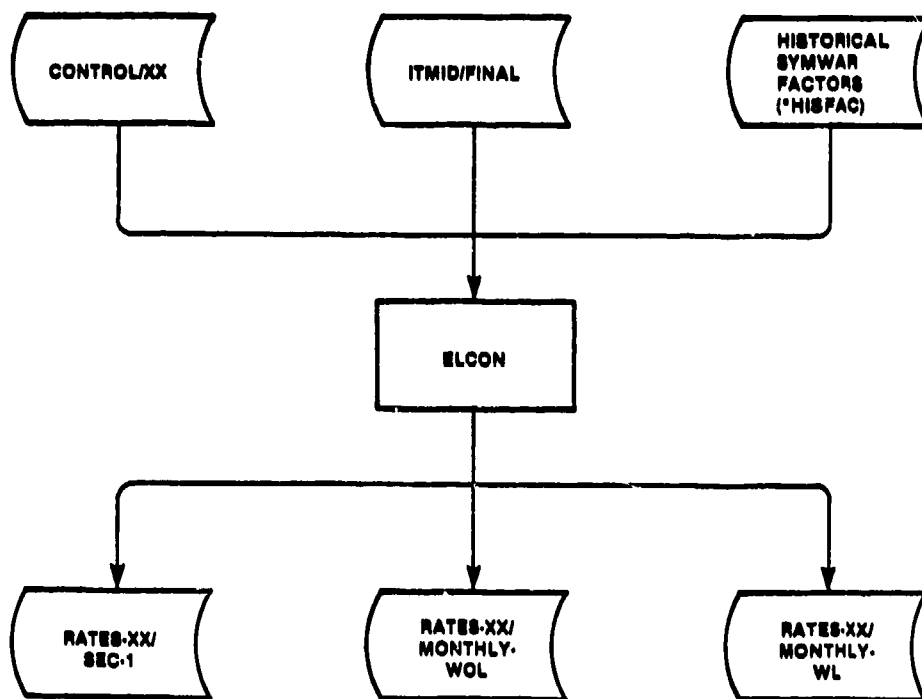


Figure III.1.1

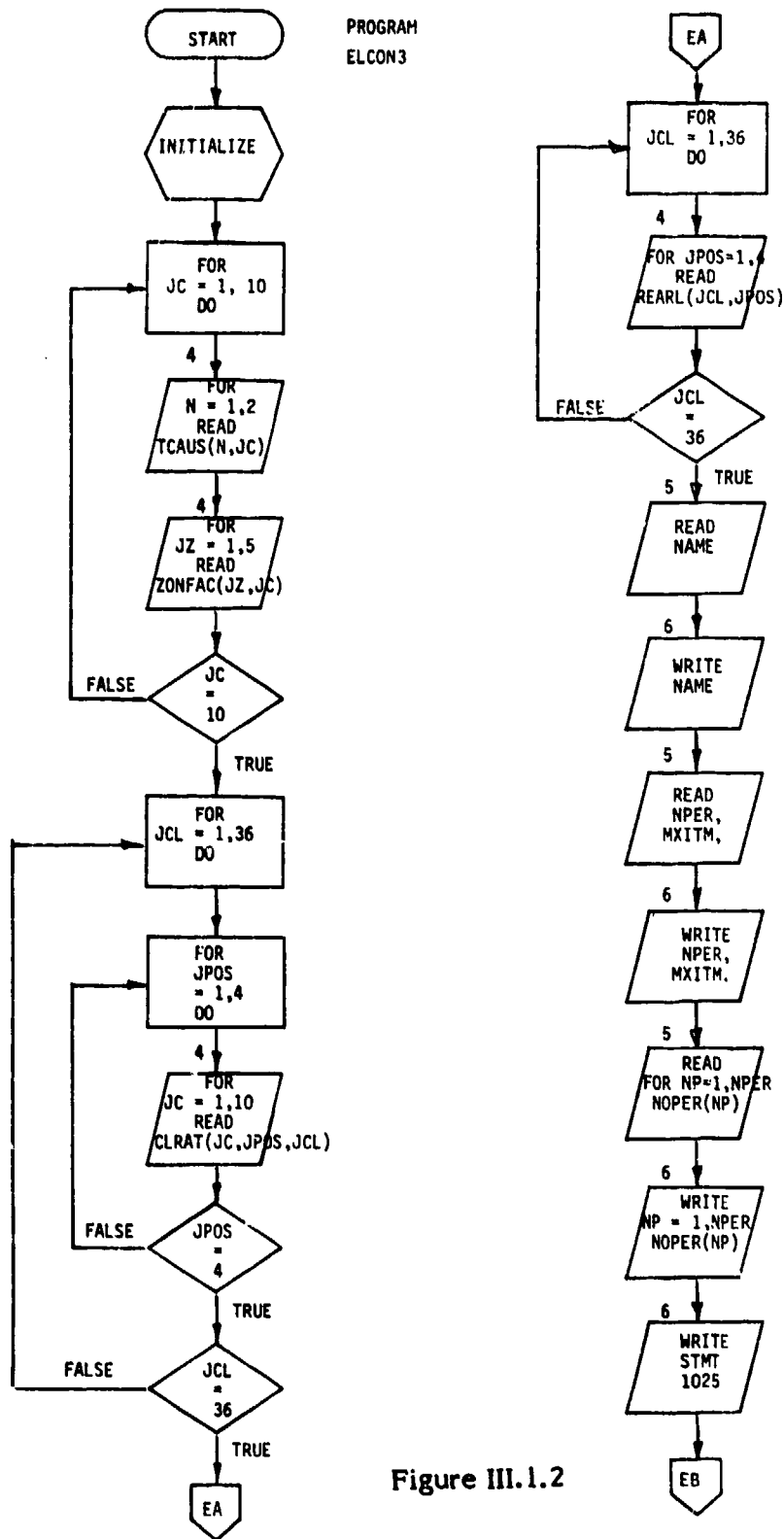


Figure III.1.2

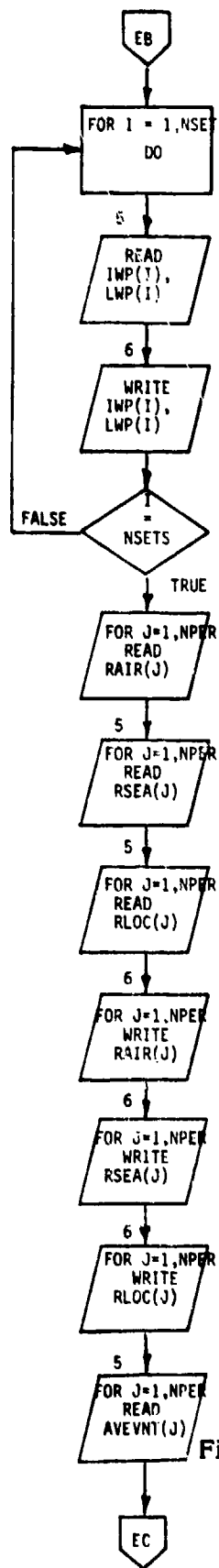
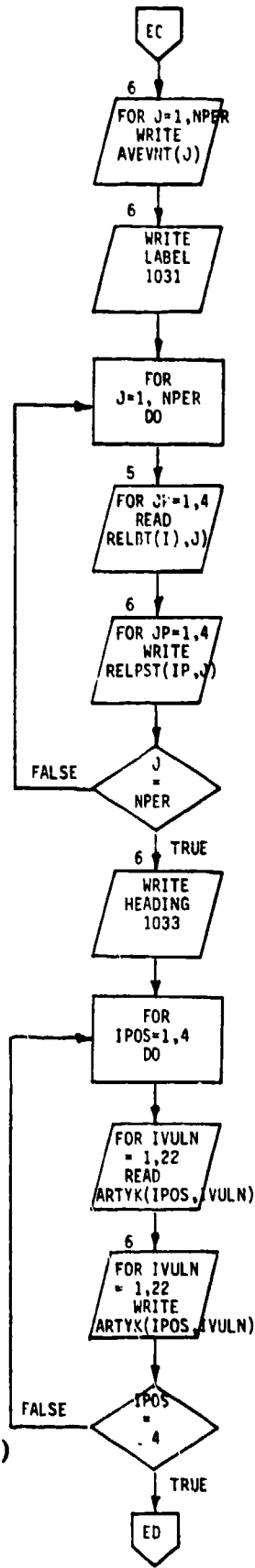


Figure III.1.2 (Cont)



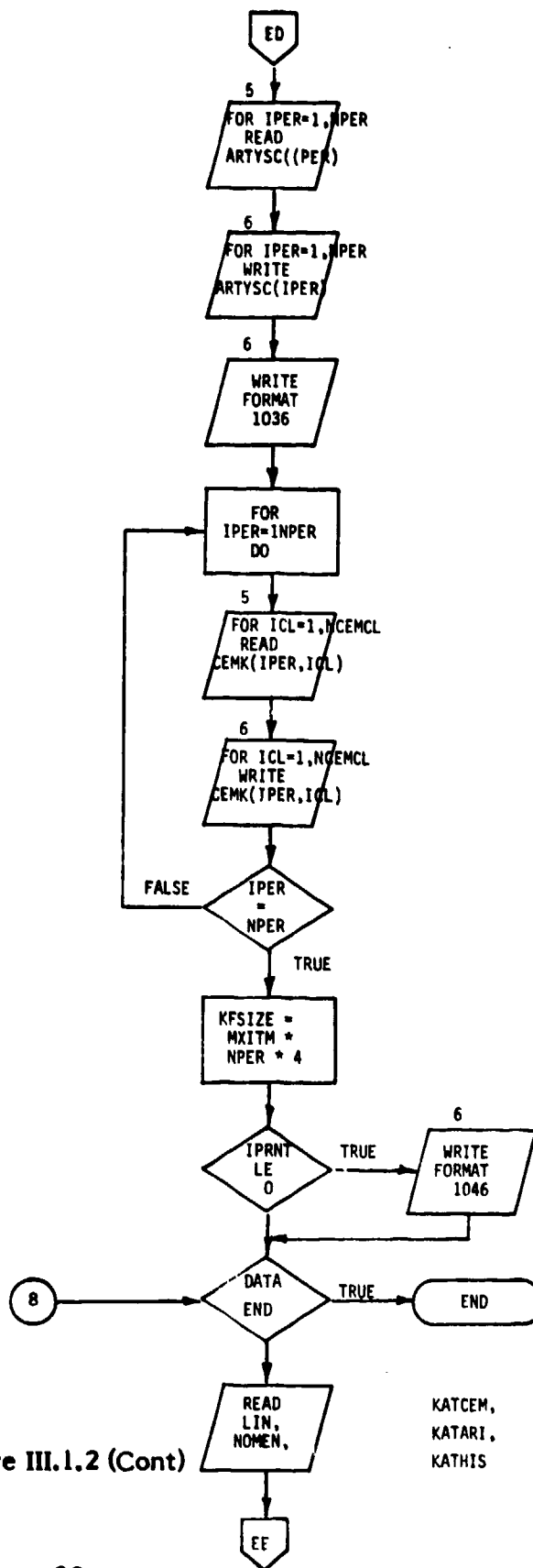


Figure III.1.2 (Cont)

KATCEM,  
KATARI,  
KATHIS

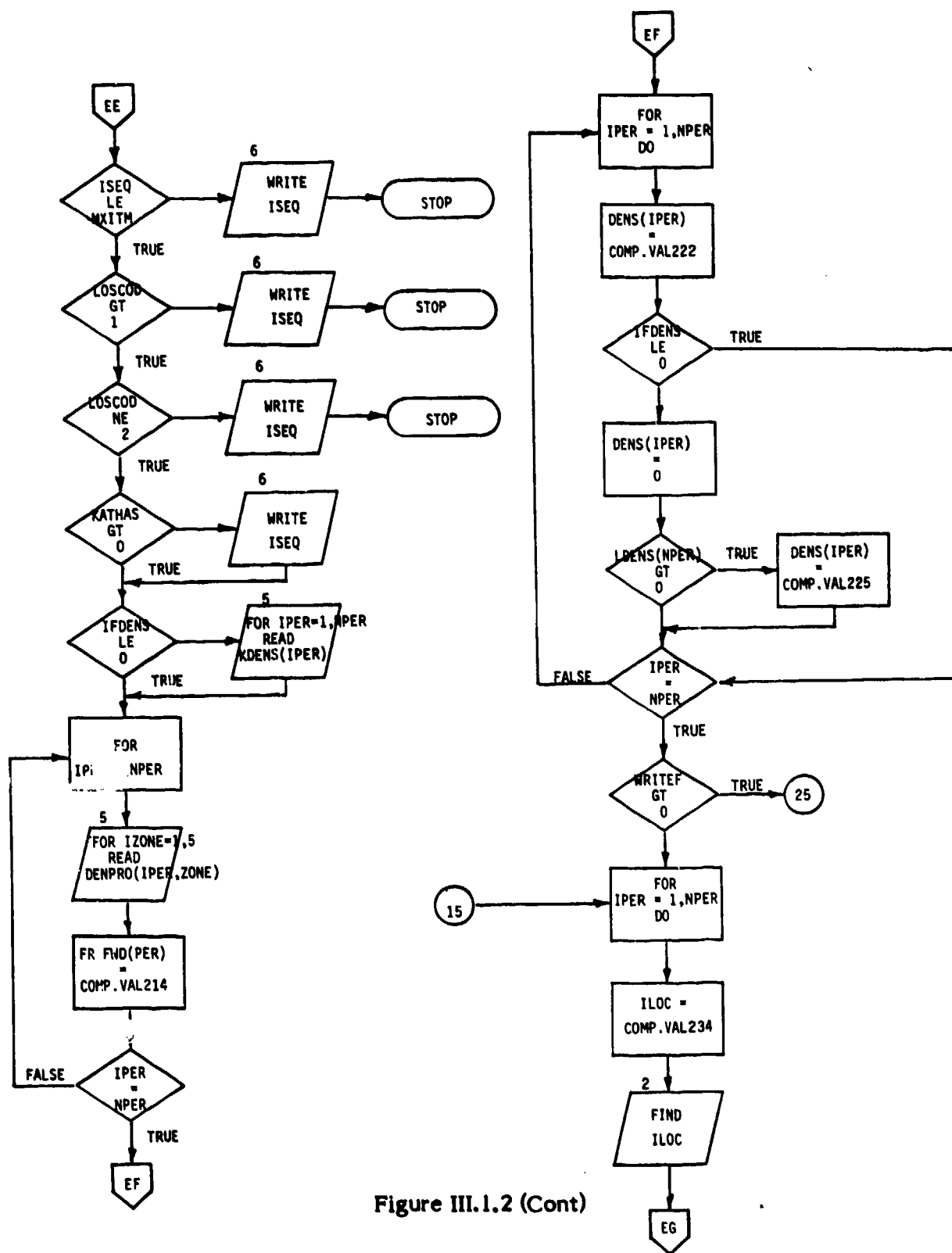
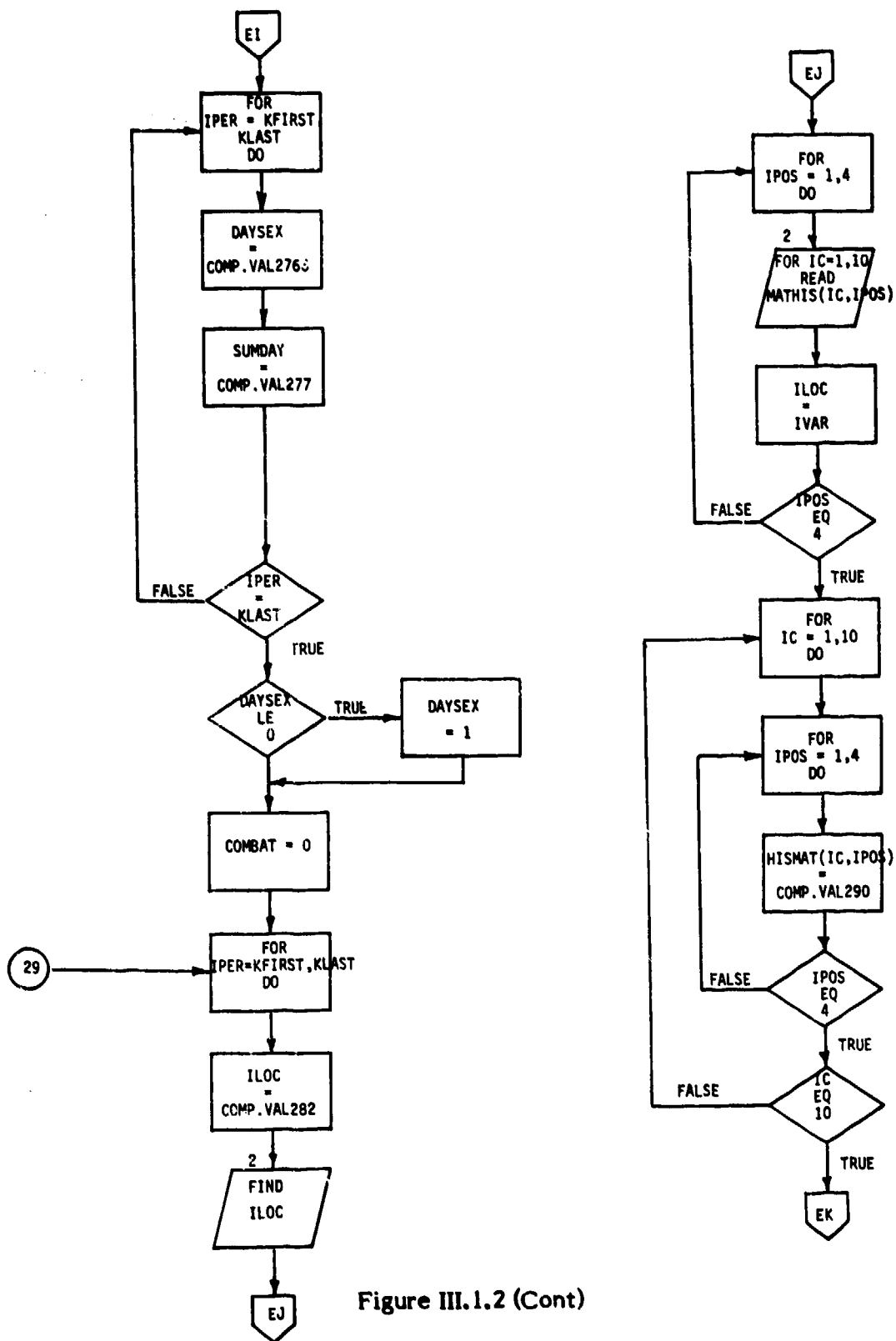


Figure III.1.2 (Cont)





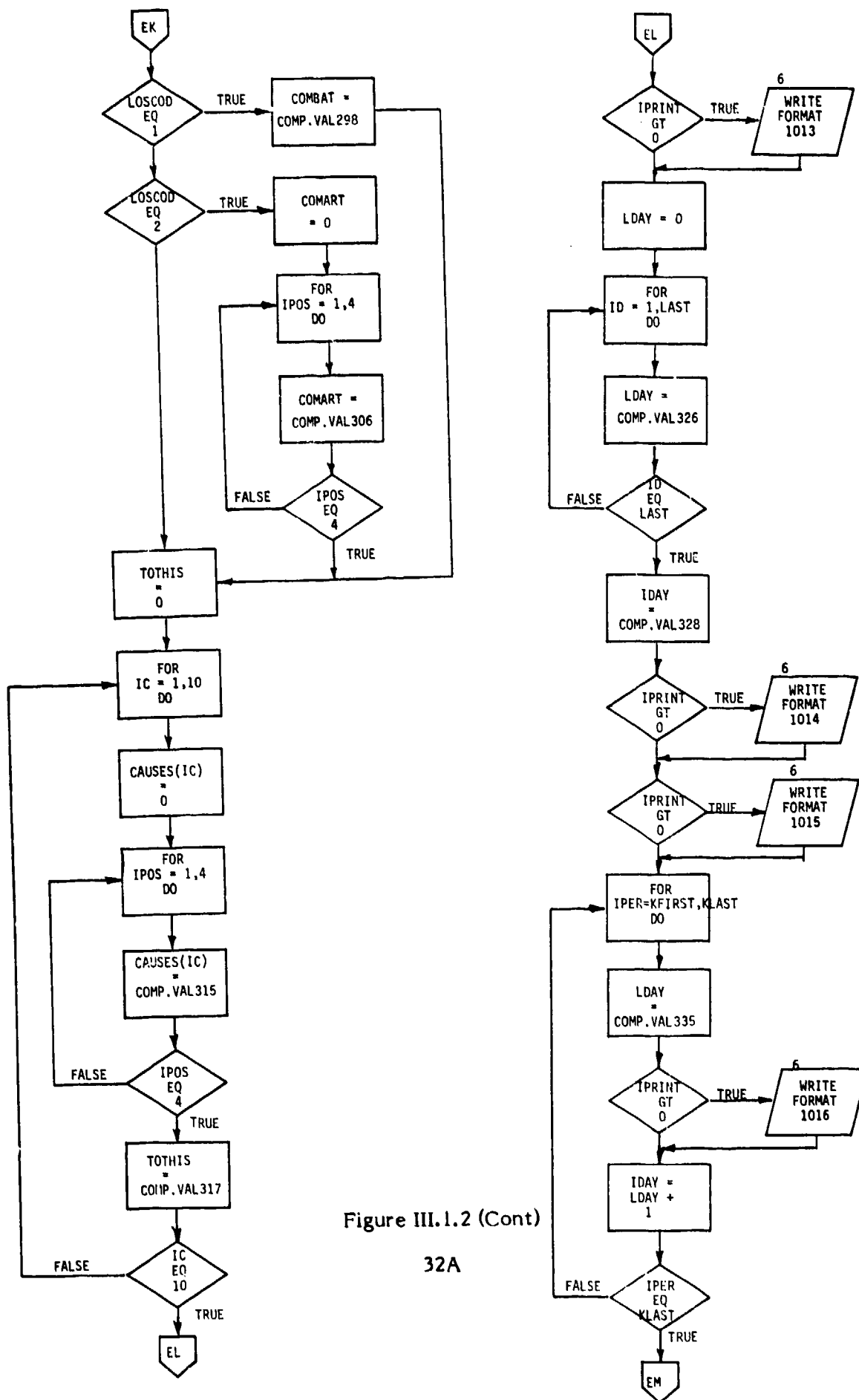
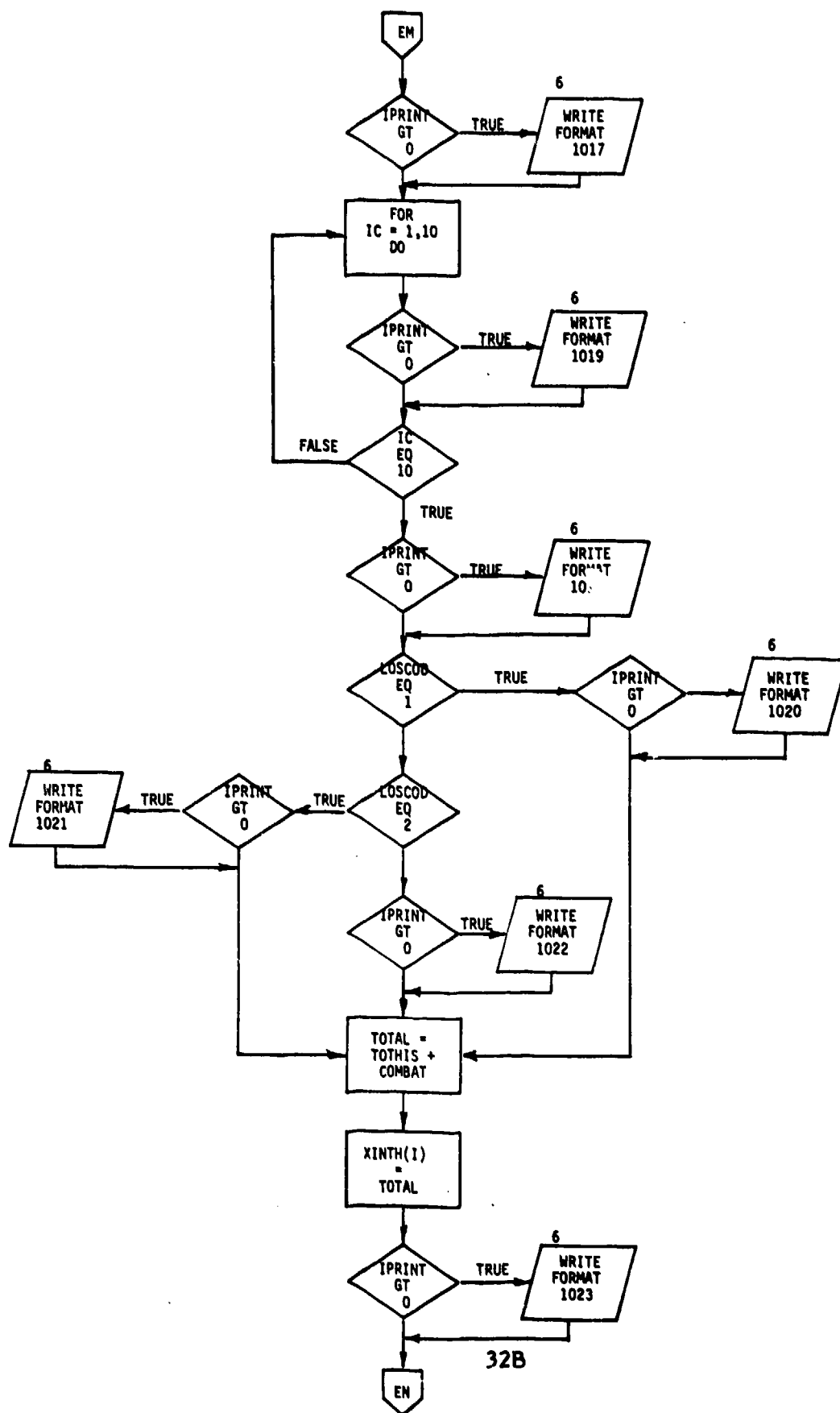


Figure III.1.2 (Cont)

32A



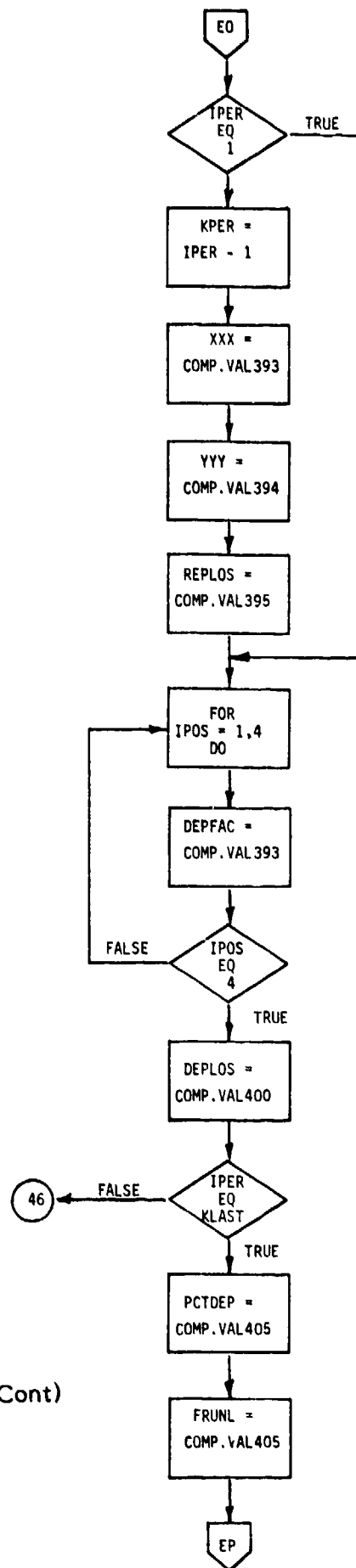
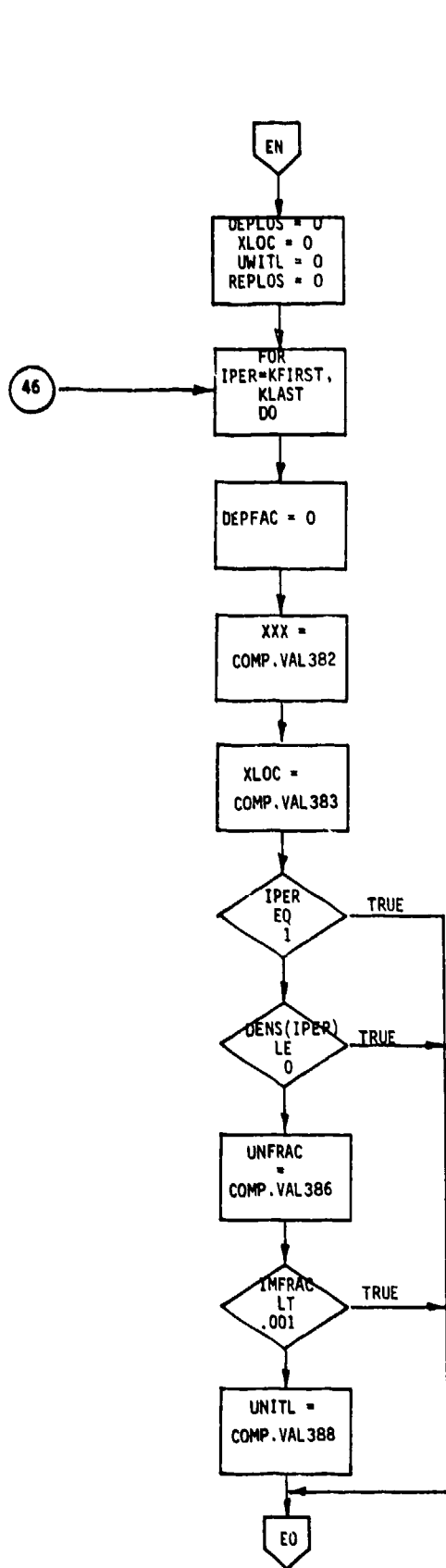


Figure III.1.2 (Cont)

THIS PAGE IS INTENTIONALLY LEFT BLANK

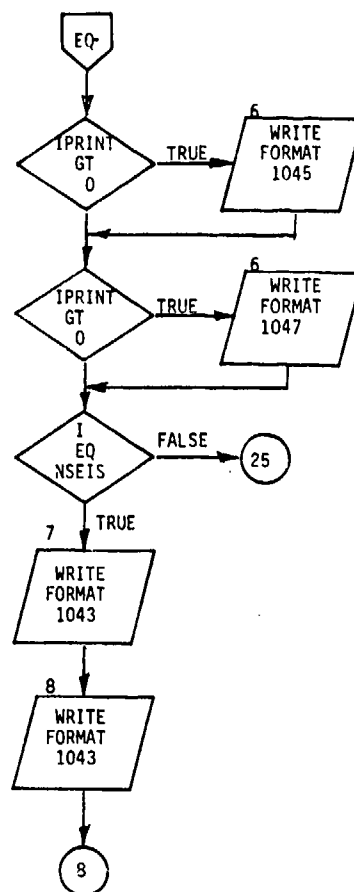
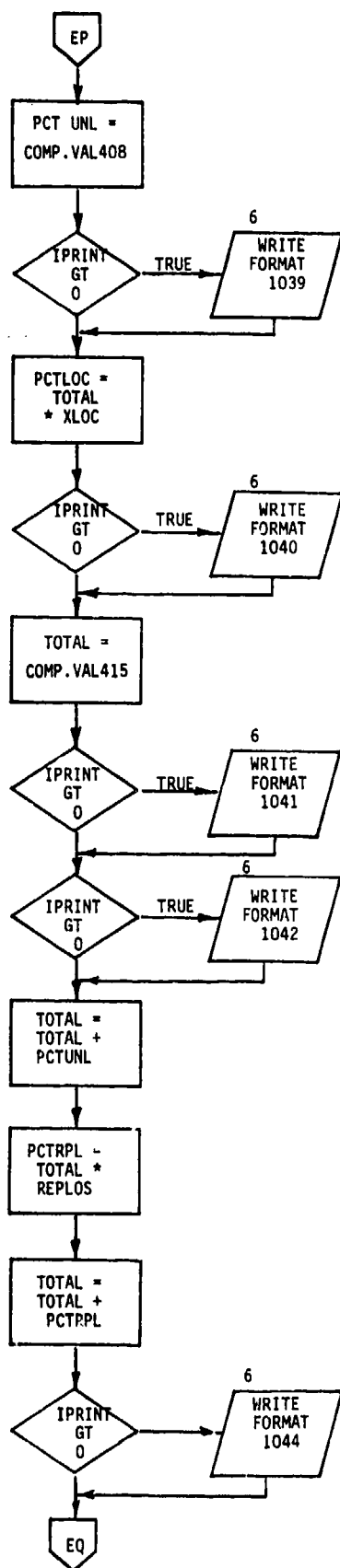


Figure III.1.2 (Cont)

UNCLASSIFIED\*\*\*FILE NAME:02XQT ELEMENT NAME:ELCON/3\*\*\*UNCLASSIFIED

```

1:C*****
2:C
3:C          E L C O N - EQUIPMENT LOSS CONSOLIDATOR
4:C
5:C*****
6:C
7:C          *****DEFINITION OF VARIABLES *****
8:C
9:C - TCAUS - ALPHANUMERIC CAUSE OF LOSS.
10:C - ZONFAC(Z,C) - THEATER AREA WEIGHT FOR ZONE Z AND CAUSE C.
11:C - CLRAT(C,P,CL) - HISTORIC (SYMMWAR) LOSS RATE FOR CAUSE C, POSTURE
12:C   P, AND HISTORIC DATA CLASSIFICATION CL.
13:C - REARL(C,P) - LOSSES IN REAR AREA (ZONES 4 AND 5) FOR HISTORICAL
14:C   CLASS C IN PERIOD P. USED IN CALCULATION OF LOSSES
15:C   OF ITEMS IN DEPOTS - DOES NOT INCLUDE LOSSES FROM
16:C   MINES, WEAROUT, OR ACCIDENT.
17:C - NAME - RUN IDENTIFIER (UP TO 30 CHARACTERS)
18:C - NPER - # OF DISTINCT TIME PERIODS
19:C - MXITH - MAXIMUM NUMBER OF EQUIPMENT ITEMS CONSIDERED.
20:C - NPER(P) - # OF DAYS IN TIME PERIOD P.
21:C - NCEMCL - # OF CEM CATEGORIES.
22:C - NSETS - # OF SETS OF WARF RATES TO BE COMPUTED.
23:C - IWP(S) - FIRST TIME PERIOD IN WARF SET S.
24:C - LWP(S) - LAST TIME PERIOD IN WARF SET S.
25:C - RAIR(P) - INTER-THEATER LOSS RATE FOR EQUIPMENT SHIPPED BY
26:C   AIR IN PERIOD P.
27:C - RSEA(P) - INTER-THEATER LOSS RATE FOR EQUIPMENT SHIPPED BY
28:C   SEA IN PERIOD P.
29:C - RLOC(P) - LOSS RATE OF REPLACEMENT EQUIPMENT ON THE IN-THEATER
30:C   LOC IN PERIOD P.
31:C - AVEUNT(P) - AVERAGE # OF MAJOR BLUE UNITS IN THEATER IN PERIOD P.
32:C - RELPST(POS,PER) - FRACTION OF TIME BLUE FORCES SPENT IN POSTURE
33:C   POS DURING PERIOD PER.
34:C - ARTYK(P,V) - FRACTION OF ARTY VULNERABILITY CATEGORY V LOST IN
35:C   ONE DAY OF POSTURE P.
36:C - ARTYSC(P) - SCALING FACTOR FOR SIMULATED ARTY KILLS FOR POSTURE P.
37:C - CEMK(P,C) - LOSSES OF CEM CATEGORY C IN PERIOD P (EXPRESSED AS
38:C   PERCENT PER 30 DAYS.
39:C - LIN - LINE ITEM NUMBER.
40:C - NOMEN - 30 CHARACTER NAME OF ITEM OF EQUIPMENT.
41:C - KATCEM - CEM CATEGORY.
42:C - KATART - ARTILLERY VULNERABILITY CLASS.
43:C - KATHIS - HISTORICAL DATA CLASS.
44:C - NDEDEF - # OF DAYS OF SUPPLY KEPT IN IN-THEATER DEPOTS.
45:C - AIRSHP - FRACTION OF RESUPPLY SHIPPED BY AIR.
46:C - IFDENS - 1, IF ACTUAL DENSITY IS TO BE USED IN CALCULATIONS.
47:C   0, IF DENSITY IS PROPORTIONAL TO MAJOR UNITS.
48:C - LOSCOD - 1, IF COMBAT LOSSES ARE FROM CEM
49:C   2, IF COMBAT LOSSES ARE FROM ARTILLERY MODEL
50:C   3, IF COMBAT LOSSES ARE FROM HISTORY.
51:C - ISEQ - ITEM SEQUENCE NUMBER.
52:C - KDENS(P) - ACTUAL DENSITY OF ITEM IN PERIOD P.
53:C - DENPRO(P,Z) - FRACTION OF DENSITY WHICH IS IN ZONE Z DURING
54:C   PERIOD P.
55:C - CAUZON(Z,C) - CAUSE BY ZONE WEIGHTS (FROM SYMMWAR).
56:C - SHZFAC(C) - CAUZON SUMMED OVER ZONES FOR CAUSE C.
57:C - DENS(P) - FRACTION OF PEAK DENSITY IN THEATER DURING PERIOD P

```

Figure III.1.3

UNCLASSIFIED\*\*\*FILE NAME:82X0T ELEMENT NAME:ELCON/3\*\*\*UNCLASSIFIED

```

58:C      TIMES 10000 (SINCE SYMWAR LOSSES ARE PER 10000 ITEMS).
59:C - WRITEF - 0, IF LOSS MATRIX HAS NOT ALREADY BEEN CREATED.
60:C      1, IF PRE-CREATED MATRIX (FILE 2) IS TO BE USED.
61:C - MATHIS(C,P), MATRIX OF HISTORICAL LOSSES FOR CAUSE C IN
62:C      POSTURE P (READY FOR ADDITION OF LOSSES FROM SIMULATIONS).
63:C      EXPRESSED AS LOSSES PER 10000 PER 30 DAYS.
64:C - HISHAT(C,P) - MATRIX OF HISTORICAL LOSSES FOR CAUSE C IN
65:C      POSTURE P FOR A PARTICULAR SET OF WARF'S (SUMMED OVER ALL
66:C      PERIODS CONSIDERED IN A WARF SET. EXPRESSED AS FRACTION
67:C      OF THOSE EXPOSED WHICH ARE LOST PER 30 DAYS.
68:C - DAYSEX - ITEM DAYS OF EXPOSURE.
69:C - SUMDAY - TOTAL DAYS CONSIDERED IN A PARTICULAR SET OF WARF'S.
70:C - COMBAT - LOSSES FROM SIMULATION (PCT PER 30 DAYS).
71:C - CAUSES(C) - HISTORICAL LOSSES FROM CAUSE C (PCT PER 30 DAYS).
72:C - TOTHS - TOTAL HISTORICAL LOSSES (ALL CAUSES).
73:C - IPRINT - 0, IF SHORT PRINT IS DESIRED.
74:C      1, IF COMPLETE LONG PRINT (1 PAGE PER WARF SET PER LIN).
75:C - XINTHS - TOTAL IN-THEATER LOSS RATE FOR WARF SET S.
76:C - XWSHP(S) - TOTAL INCLUDING INTER-THEATER SHIPPING LOSSES.
77:C - DEPOS - LOSS RATE OF EQUIPMENT IN DEPOTS.
78:C - XLOC - LOSS RATE DURING IN-THEATER TRANSIT.
79:C - UNFRAC - FRACTION OF CURRENT THEATER STRENGTH WHICH IS CURRENTLY
80:C      IN TRANSIT TO THE THEATER.
81:C - UNITL - LOSS RATE DURING INTER-THEATER TRANSIT OF REINFORCING
82:C      UNITS.
83:C - REPLOS - LOSS RATE DURING INTER-THEATER TRANSIT OF REPLACEMENT
84:C      EQUIPMENT.
85:C
86:C      *****BEGINNING OF CODE*****
87:      DIMENSION TCAUS(2,10),ZCNFAC(5,10),CLRAT(10,4,36),NAME(12),
88:      1  NOPER(20),IMP(20),LWF(20),RAIR(20),RSEA(20),RLOC(20),
89:      2  AVEUNT(20),RELPS(4,20),ARTYK(4,22),ARTYSC(20),CEMK(20,25)
90:      DIMENSION NOMEN(5),KDENS(20),CAUZON(5,10),SMZFAC(10),
91:      1  MATHIS(10,4),HISHAT(10,4),CAUSEC(10),DFNPPG(20,5),
92:      2  DENS(20),FRFWD(20),REARL(36,4),XINTH(20),XWSHP(20)
93:      INTEGER WRITEF
94:      IHIS=4
95:      IN=5
96:      IO=6
97:      INOSHP=7
98:      IWSHP=8
99:C
100:C - READ SYMWAR HISTORICAL RATE FACTORS.
101:C
102:      DO 1 JC=1,10
103:      READ(IHIS,1001)(TCAUS(N,JC),N=1,2),(ZCNFAC(JZ,JC),JZ=1,5)
104:1001  FORMAT(4X,2A6,9X,5F6.2)
105:1    CONTINUE
106:      DO 2 JCL=1,36
107:      DO 3 JPOS=1,4
108:      READ(IHIS,1002)(CLRAT(JC,JPOS,JCL),JC=1,10)
109:1002  FORMAT(10F8.5,4X)
110:3    CONTINUE
111:2    CONTINUE
112:      DO 45 JCL=1,36
113:      READ(IHIS,1038)(REARL(JCL,JPOS),JPOS=1,4)
114:1038  FORMAT(4F12.6)

```

Figure III.1.3 (Cont)

UNCLASSIFIED\*\*\*FILE NAME:82XQT ELEMENT NAME:ELCON/3\*\*\*UNCLASSIFIED

```

115:45    CONTINUE
116:C
117:C - READ INPUTS FOR THIS RUN.
118:C
119:      READ(IN,1003)NAME
120:1003  FORMAT(12A6)
121:      WRITE(IO,1004)NAME
122:1004  FORMAT(1H1,12A6)
123:      READ(IN,1005)NPER,MXITM,NCEMCL,NSETS,WRITEF,NPPRES,IPRNT
124:      WRITE(IO,1024)NPER,MXITM,NCEMCL,NSETS
125:1024  FORMAT(////' NPER MXITM # CEM CLASSES #WARF SETS'/
126:      1  IS,I7,6X,I4,7X,I7)
127:1005  FORMAT(16I5)
128:      READ(IN,1005)(NPER(NP),NP=1,NPER)
129:      WRITE(IO,1025)(NPER(NP),NP=1,NPER)
130:1025  FORMAT(//' DAYS PER PERIOD'/10I5)
131:      WRITE(IO,1026)
132:1026  FORMAT(//' FIRST AND LAST PERIOD FOR EACH WARF SET')
133:      DO 4 I=1,NSETS
134:      READ(IN,1005)IWP(I),LWP(I)
135:      WRITE(IO,1005)IWP(I),LWP(I)
136:4      CONTINUE
137:      READ(IN,1006)(RAIR(J),J=1,NPER)
138:      READ(IN,1006)(RSEA(J),J=1,NPER)
139:      READ(IN,1006)(RLOC(J),J=1,NPER)
140:      WRITE(IO,1027)(RAIR(J),J=1,NPER)
141:1027  FORMAT(//' AIR LOSS RATE BY PERIOD'/10F7.3)
142:      WRITE(IO,1028)(RSEA(J),J=1,NPER)
143:1028  FORMAT(//' SEA LOSS RATE BY PERIOD'/10F7.3)
144:      WRITE(IO,1029)(RLOC(J),J=1,NPER)
145:1029  FORMAT(//' LOC LOSS RATE BY PERIOD'/10F7.3)
146:1006  FORMAT(16F5.4)
147:      READ(IN,1007)(AVEUNT(J),J=1,NPER)
148:      WRITE(IO,1030)(AVEUNT(J),J=1,NPER)
149:1030  FORMAT(//' AVERAGE LARGE UNITS BY PERIOD'/10F7.2)
150:1007  FORMAT(16F5.2)
151:      WRITE(IO,1031)
152:1031  FORMAT(//' FRACTION OF FORCE BY POSTURE BY PERIOD'/
153:      1  ' ATTACK DEFEND WITHDRAW INACTIVE')
154:      DO 5 J=1,NPER
155:      READ(IN,1007)(RELPS(ITF,J),IP=1,4)
156:      WRITE(IO,1032)(RELPS(ITF,J),IP=1,4)
157:1032  FORMAT(2F7.4,2F10.6)
158:5      CONTINUE
159:      WRITE(IO,1033)
160:1033  FORMAT(//' ARTY KILLS ( : PER DAY) BY VULN CLASS BY POSTURE')
161:      DO 6 IPOS=1,4
162:      READ(IN,1001)(ARTYK(IPOS,IVULN),IVULN=1,22)
163:2001  FORMAT(11F6.3)
164:      WRITE(IO,1034)(ARTYK(IPOS,IVULN),IVULN=1,22)
165:1034  FORMAT(11F8.3/11F8.3/)
166:6      CONTINUE
167:      READ(IN,150)(ARTYSC(IPFR),IPFR=1,NPER)
168:156  FORMAT(1X,16F5.3)
169:      WRITE(IO,1035)(ARTYSC(IPFR),IPFR=1,NPER)
170:1035  FORMAT(//' ARTY SCALNG FACTORS BY PERIOD'/1X,10F7.3)
171:      WRITE(IO,1036)

```

Figure III.1.3 (Cont)

UNCLASSIFIED\*\*\*FILE NAME:82XQT ELEMENT NAME:ELCON/3\*\*\*UNCLASSIFIED

```

172:1036 FORMAT(/** CEM KILLS (2 PER 30 DAYS) BY CLASS BY PERIOD*)
173:      DO 7 IPER=1,NPER
174:      READ(IN,155)(CEMK(IPER,ICL),ICL=1,NCEMCL)
175:155   FORMAT(1X,13F6.2)
176:      WRITE(IO,1037)(CEMK(IPER,ICL),ICL=1,NCEMCL)
177:1037  FORMAT(1X,10F7.3)
178:17   CONTINUE
179:      KFSIZE=MXITM*NPER*4
180:      DEFINE FILE 2(KFSIZE,10,U,IVAR)
181:C     IF(IFRNT.LE.0)WRITE(IC,1046)
182:1046  FORMAT('1 ITEM OF EQUIPMENT',1EX,'WAF IN-THEATER',3X,
183:      1 'WITH INTER-'/4DX,'SET LOSSES THEATER LOSSES'/)
184:C
185:C - BEGIN EQUIPMENT ITEM PROCESSING.
186:C
187:18   CONTINUE
188:      READ(IN,1008,END=100)LIN,NOMEN,KATCEM,KATART,KATHIS.
189:      1 NDCP,ATRSHP,IFDENS,LCSOD,ISEG
190:1008  FORMAT(1X,A6,1X,5A6,1X,3I2,I3,F2.2,2I2,I4)
191:C
192:C - CHECK FOR OBVIOUS INPUT ERRORS.
193:C
194:      IF(ISEG.LE.MXITM) GO TO 9
195:      WRITE(IO,1009)ISEG
196:1009  FORMAT(' SEQUENCE #',I5,' LARGER THAN MAX SPECIFIED')
197:      STOP
198:19   IF(LOSOD.GT.1.OR.KATCEM.GT.0)GO TO 10
199:      WRITE(IO,1010)ISEG
200:1010  FORMAT(' SEQUENCE #',I5,' MARKED AS CEM ITEM W/O CATCEM')
201:      STOP
202:10   IF(LOSOD.NE.2.OR.KATART.GT.0) GO TO 11
203:      WRITE(IO,1011)ISEG
204:1011  FORMAT(' SEQUENCE #',I5,' MARKED AS ARTY ITEM W/C CATART')
205:      STOP
206:11   IF(KATHIS.GT.0.AND.KATHIS.LE.36) GO TO 12
207:      WRITE(IO,1012)ISEG
208:1012  FORMAT(' ITEM #',I5,' HAS ILLEGAL HISTORICAL CLASS')
209:12   IF(IFDENS.LE.0)GO TO 13
210:      READ(IN,2005)(KDENS(IPER),IPER=1,NPER)
211:2005  FORMAT(10I7)
212:13   DO 14 IPER=1,NPER
213:      READ(IN,1007)(DENPRO(IPER,IZONE),IZONE=1,5)
214:      FRWD(IPER)=DENPRO(IPER,1)+DENPRO(IPER,2)+
215:      1 DENPRO(IPER,3)
216:14   CONTINUE
217:C
218:C - COMPUTE EACH PERIOD'S FRACTION OF PEAK THEATER DENSITY
219:C OF THIS ITEM.
220:C
221:      DO 27 IPER=1,NPER
222:      DENS(IPER)=AVEUNT(IPER)*10000/AVEUNT(NPER)
223:      IF(IFDENS.LE.0)GO TO 27
224:      DENS(IPER)=0.
225:      IF(KDENS(IPER).GT.0)DENS(IPER)=KDENS(IPER)*10000/KDENS(NPER)
226:27   CONTINUE
227:C
228:C - IF LOSS MATRIX FOR EACH PERIOD HAS NOT ALREADY BEEN

```

Figure III.1.3 (Cont)

UNCLASSIFIED\*\*FILE NAME:82X07 ELEMENT NAME:ELCON/3\*\*UNCLASSIFIED

```

229:C CONSTRUCTED AND INPUT ON FILE 2. CONSTRUCT THE MATRIX
230:C AND STORE ON FILE 2.
231:C
232: IF(WRITEF.GT.0)GO TO 44
233: DO 15 IPER=1,NPER
234: ILOC=(ISEQ-1)*NPER*4 + (IPER-1)*4 + 1
235: FIND(2*ILOC)
236: IF(IPER.EQ.1) GO TO 16
237: ITEST=IPER-1
238: DO 17 I=1,5
239: IF (IDENPRO(IPER,I).NE.DENPRO(ITEST,I))GO TO 16
240:17 CONTINUE
241: GO TO 18
242:16 CONTINUE
243: DO 22 IC=1,10
244: SMZFAC(IC)=0.
245: DO 19 IZ=1,5
246: IF(LOSCOD.GT.2)GO TO 20
247: IF(LOSCOD.EQ.1.AND.IC.EQ.1.AND.IZ.LE.3)GO TO 19
248: GO TO(20,21,20,21,21,20,20,20,20),IC
249:21 GO TO(19,19,19,20,20),IZ
250:20 CAUZON(IZ,IC)=ZONFAC(IZ,IC)*DENPRO(IPER,IZ)
251: SMZFAC(IC)=SMZFAC(IC)+CAUZON(IZ,IC)
252:19 CONTINUE
253: DO 23 IPOS=1,4
254: MATHIS(IC,IPOS)=CLRAT(IC,IPOS,KATHIS)*SMZFAC(IC)*.5
255:23 CONTINUE
256:22 CONTINUE
257:18 DO 24 IPOS=1,4
258: WRITE(2*ILOC)(MATHIS(IC,IPOS),IC=1,10)
259: ILOC=IVAR
260:24 CONTINUE
261:15 CONTINUE
262:44 CONTINUE
263:C
264:C - BEGIN LOOP TO COMPUTE EACH SET OF WARE RATES
265:C FOR THIS ITEM.
266:C
267: DO 25 I=1,NSETS
268: DO 28 IC=1,10
269: DO 28 IPOS=1,4
270:28 HISMAT(IC,IPOS)=0.
271: KFIRST=LWP(I)
272: KLAST=LWP(I)
273: DAYSEX=0.
274: SUMDAY=0.
275: DO 26 IPER=KFIRST,KLAST
276: DAYSEX=DAYSEX+DENS(IPER)*NDPER(IPER)
277: SUMDAY=SUMDAY+NDPER(IPER)
278:26 CONTINUE
279: IF(DAYSEX.LE.0.)DAYSEX=1.
280: COMBAT=0.
281: DO 29 IPER=KFIRST,KLAST
282: ILOC=(ISEQ-1)*NPER*4+(IPER-1)*4+1
283: FIND(2*ILOC)
284: DO 31 IPOS=1,4
285: READ(2*ILOC)(MATHIS(IC,IPOS),IC=1,10)

```

Figure III.1.3 (Cont)

UNCLASSIFIED\*\*FILE NAME:82XQT ELEMENT NAME:ELCON/3\*\*UNCLASSIFIED

```

286:      ILOC=IVAR
287:31    CONTINUE
288:      DO 30 IC=1,10
289:      DO 30 IPOS=1,4
290:      HISMAT(IC,IPOS)=HISMAT(IC,IPOS)+(MATHIS(IC,IPOS)*
291:      1  DENS(IPER)*NDPER(IPER)*RELPST(IPOS,IPER))/
292:      2  (10000.*DAYSEX)
293:30    CONTINUE
294:      GO TO(37,38,29),LOSCOD
295:C
296:C - COMPUTE COMBAT LOSSES FOR CEM ITEMS.
297:C
298:37    COMBAT=COMBAT+CEMK(IPER,KATCEM)*FRFWD(IPER)*DENS(IPER)*
299:      1  NDPER(IPER)/DAYSEX
300:      GO TO 29
301:38    COMART=0.
302:C
303:C - COMPUTE COMBAT LOSSES FOR ARTILLERY ITEMS.
304:C
305:      DO 39 IPOS=1,4
306:      COMART=COMART+ARTYK(IPOS,KATART)*RELPST(IPOS,IPER)
307:39    CONTINUE
308:      COMBAT=COMBAT+30.*COMART*FRFWD(IPER)*ARTYSC(IPER)*
309:      1  NDPER(IPER)*DENS(IPER)/DAYSEX
310:29    CONTINUE
311:      TOTHS=0.
312:      DO 32 IC=1,10
313:      CAUSES(IC)=0.
314:      DO 33 IPOS=1,4
315:      CAUSES(IC)=CAUSES(IC)+HISMAT(IC,IPOS)*100
316:33    CONTINUE
317:      TOTHS=TOTHS+CAUSES(IC)
318:32    CONTINUE
319:      IF(IPRNT.GT.0)
320:      1  WRITE(IO,1013)LIN,NCHEN,KATCEM,KATART,KATHIS,LOSCOD,
321:      2  ISEQ
322:1013  FORMAT(1H1,1X,A6,1X,5A6,2X,'CEM=',I3,' APTY=',I3,
323:      1  ' HISTORY=',I3,4X,'SOURCE=',I2,4X,'SEQ',I5)
324:      LDAY=0
325:      DO 34 ID=1,KLAST
326:      LDAY=LDAY+NDPER(ID)
327:34    CONTINUE
328:      IDAY=LDAY-SUMDAY+1
329:      IF(IPRNT.GT.0)WRITE(IO,1014)IDAY,LDAY
330:1014  FORMAT(///3X,'30DAY LOSS RATE FOR PERIOD FROM DAY',
331:      1  I4,' TO DAY',I4)
332:      IF(IPRNT.GT.0)WRITE(IO,1015)
333:1015  FORMAT(//15X,'DENSITY PROFILE (FRACTION IN ZONES 1-5)')
334:      DO 35 IPER=KFIRST,KLAST
335:      LDAY=IDAY+NDPER(IPER)-1
336:      IF(IPRNT.GT.0)WRITE(IO,1016)IDAY,LDAY,(DENPRO(IPER,I2),I2=1,5)
337:1016  FORMAT(/19X,'DAY',I4,' - ',I4,5(3X,F4.2))
338:      IDAY=LDAY+1
339:35    CONTINUE
340:      IF(IPRNT.GT.0)WRITE(IO,1017)
341:1017  FORMAT(/10X,'RATES DERIVED FROM HISTORY (% PER 30 DAYS)')
342:      DO 36 IC=1,10

```

Figure III.1.3 (Cont)

UNCLASSIFIED\*\*\*FILE NAME:82X0T ELEMENT NAME:ELCON/T\*\*\*UNCLASSIFIED

```

343:      IF(IPRNT.GT.0)WRITE(10,1018)(TCAUS(N,IC),N=1,2),CAUSES(IC)
344:1018  FORMAT(13X,2A6,F9.2)
345:36    CONTINUE
346:      IF(IPRNT.GT.0)WRITE(10,1019)TOTHIS
347:1019  FORMAT(//10X,'TOTAL FROM HISTORY = ',F7.2)
348:      GO TO(40,41,42),LOSCOD
349:40     IF(IPRNT.GT.0)WRITE(10,1020)COMBAT
350:1020  FORMAT(//10X,'ADDITIONAL RATES DERIVED BY SIMULATION.'
351:      1 1X,'(INCLUDES DIRECT FIRE, AREA FIRE, AND AIR)'/
352:      2 27X,F6.2)
353:      GO TO 43
354:41     IF(IPRNT.GT.0)WRITE(10,1021)COMBAT
355:1021  FORMAT(//10X,'ADDITIONAL RATES DERIVED BY SIMULATION'
356:      1 1X,'(INCLUDES AREA FIRE AND AIR)'/27X,F6.2)
357:      GO TO 43
358:42     IF(IPRNT.GT.0)WRITE(10,1022)
359:1022  FORMAT(//10X,'NO ADDITIONAL RATES FROM SIMULATION')
360:43     TOTAL=TOTHIS+COMBAT
361:      XINTH(1)=TOTAL
362:      IF(IPRNT.GT.0)WRITE(10,1023)TOTAL
363:1023  FORMAT(//10X,'TOTAL LOSS RATE (NOT INCLUDING SHIPPING ',
364:      1 'OR DEPOT LOSSES)'/27X,F6.2)
365:C
366:C - COMPUTE ADDITIONAL LOSSES (DEPOT,LOC,INTER-THEATER LOSSES IN
367:C   UNITS, INTER-THEATER LOSSES OF REPLACEMENT EQUIPMENT).
368:C
369:      DEPOS=0.
370:      XLOC=0.
371:      UNITL=0.
372:      REPLOS=0.
373:      DO 46 IPER=KFIRST,KLAST
374:      DEPFAC=0.
375:C
376:C *****
377:C   THOSE STATEMENTS IN THE NEXT SECTION OF CODE WHICH ARE MARKED
378:C   WITH A "1" IN COLUMN 73 HAVE BEEN CHANGED OR ADDED SINCE THE
379:C   MODEL DOCUMENTATION PUBLISHED IN AUGUST 1979.
380:C *****
381:C
382:      XXX=RLOC(IPER)/(1.-RLOC(IPER))
383:      XLOC=XLOC+DENS(IPER)*NDPER(IPER)*XXX/DAYSEX
384:      IF(IPER.EQ.1) GO TO 48
385:      IF(DENS(IPER).LE.0.)GO TO 48
386:      UNFRAC=(DENS(IPER)-DENS(IPER-1))/DENS(IPER)
387:      IF(UNFRAC.LT..001)GO TO 48
388:      UNITL=UNITL+(RAIR(IPER-1)*AIRSHP+RSEA(IPER-1)*(1.-AIRSHP))
389:      1 *UNFRAC*DENS(IPER)
390:48     CONTINUE
391:      IF(IPER.EQ.1.OR.IPER.LE.NPPRES) GO TO 49
392:      KPER=IPER-1
393:      XXX=RAIR(KPER)/(1.-RAIR(KPER))
394:      YYY=RSEA(KPER)/(1.-RSEA(KPER))
395:      REPLOS=REPLOS+(XXX*AIRSHWP+YYY*(1.-AIRSHP))*
396:      1 DENS(IPER)*NDPER(IPER)/DAYSEX
397:49     CONTINUE
398:      DO 47 IPOS=1,*
399:47     DEPFAC=DEPFAC+REARL(KATHTS,IPOS)*RELPOST(IPOS,IPER)

```

Figure III.1.3 (Cont)

UNCLASSIFIED\*\*\*FILE NAME:82XGT ELEMENT NAME:ELCON/3\*\*\*UNCLASSIFIED

```

400:      DEPOS=DEPOS+DEPFAC*DE NS(IPER)*NPER(IPER)/DAYSEX
401:46      CONTINUE
402:C      DELETED FROM ORIGINAL
403:C      DELETED FROM ORIGINAL
404:C      DELETED FROM ORIGINAL
405:      PCTDEP=TOTAL*(NDEP/30.)*DEPOS
406:      FRUNL=UNITL*30./DAYSEX
407:C      DELETED FROM ORIGINAL
408:      PCTUNL=FRUNL*100.
409:      IF(IPRNT.GT.0)WRITE(IO,1039)DEPOS,NDEP,PCTDEP
410:1039  FORMAT(/10X,'LOSSES IN DEPOT ('F5.3,' OF',I4,
411:      1 ' DAYS STOCK) = 'F5.3,'%)
412:      PCTLOC=TOTAL*XLOC
413:      IF(IPRNT.GT.0)WRITE(IO,1040)PCTLOC
414:1040  FORMAT(/10X,'LOSSES IN IN-THEATER LOC ='F6.3,'%)
415:      TOTAL=TOTAL+PCTLOC+PCTDEP
416:      IF(IPRNT.GT.0)WRITE(IO,1041)TOTAL
417:1041  FORMAT(/10X,'TOTAL IN-THEATER LOSSES ='F7.2,'%)
418:      IF(IPRNT.GT.0)WRITE(IO,1042)PCTUNL
419:1042  FORMAT(/10X,'INTER-THEATER LOSS DURING UNIT TRANSPORT ='
420:      1 F7.3,'%)
421:C*****
422:C      THE FOLLOWING CODE REFLECTS A CHANGE IN THE
423:C      DOCUMENTATION IN THE WAY THE REPLACEMENT
424:C      FACTOR IS APPLIED IN COMPUTING A WARF.
425:C*****
426:      TOTAL=TOTAL+PCTUNL
427:      PCTRPL=TOTAL*REPOS
428:      TOTAL=TOTAL+PCTRPL
429:      IF(IPRNT.GT.0)WRITE(IO,1044)PCTRPL
430:1044  FORMAT(/10X,'INTER-THEATER LOSS OF REPLACEMENT ITEMS = ',
431:      1 F7.3,'%)
432:      IF(IPRNT.GT.0)WRITE(IO,1045)TOTAL
433:1045  FORMAT(/20X,'TOTAL 30 DAY LOSS RATE ='F7.2,'%)
434:      XWSHP(I)=TOTAL
435:      IF(IPRNT.EQ.0)WRITE(IO,1047)LIN,NOMEN,I,XINTH(I),XWSHP(I)
436:1047  FORMAT(1X,A6,1X,5A6,I4,F9.2,F13.2)
437:25      CONTINUE
438:      WRITE(INOSHP,1043)LIN,NOMEN,(XINTH(ISET),ISET=1,NSETS)
439:      WRITE(IWSHP,1043)LIN,NOMEN,(XWSHP(ISET),ISET=1,NSETS)
440:1043  FORMAT(1X,A6,1X,5A6,10F6.2/39X,10F6.2)
441:      GO TO 8
442:100      CONTINUE
443:      END

```

Figure III.1.3 (Cont)



UNCLASSIFIED\*\*\*EXAMPLE OF THE OUTPUT OF UTILITY ITHID/REC-A\*\*\*UNCLASSIFIED

1:	AD3198	AK	VEH	M218	GM	EG	P1A			U 522	30	0	1	2	1
2:	54		64		44		58	78		88		88			
3:	.00	.00	.00	.80	.20										
4:	.00	.00	.00	.90	.20										
5:	.00	.00	.00	.80	.20										
6:	.00	.00	.00	.80	.20										
7:	.00	.00	.00	.80	.20										
8:	.00	.00	.00	.80	.20										
9:	.00	.00	.00	.75	.25										
10:	A14752	ADAP	TEST	CAMERA	LM178					U1636	30	0	1	2	2
11:	10		16		16	17	17			17		17			
12:	.00	.00	.20	.90	.00										
13:	.00	.00	.20	.80	.00										
14:	.00	.00	.20	.80	.00										
15:	.00	.00	.15	.85	.00										
16:	.00	.00	.15	.85	.00										
17:	.00	.00	.35	.65	.00										
18:	.00	.00	.35	.65	.00										
19:	A2249C	AIMING	CIRCLE	M2	W/E					01636	30	0	1	2	3
20:	6615		6699		8820	8923	8823	8923	8978						
21:	.25	.25	.50	.00	.00										
22:	.25	.25	.50	.00	.00										
23:	.25	.25	.50	.00	.00										
24:	.25	.25	.50	.00	.00										
25:	.25	.25	.50	.00	.00										
26:	.25	.25	.50	.00	.00										
27:	.25	.25	.50	.00	.00										
28:	A23770	AIR	COND	FL/WNDW	60008					01833	30	0	1	2	4
29:	0		0		0	0	0			0		0			
30:	.00	.00	.00	.00	.00										
31:	.00	.00	.00	.00	.00										
32:	.00	.00	.00	.00	.00										
33:	.00	.00	.00	.00	.00										
34:	.00	.00	.00	.00	.00										
35:	.00	.00	.00	.00	.00										
36:	.00	.00	.00	.00	.00										
37:	A23828	AIR	COND	F/WA	9000	BTU				01833	30	0	1	2	5
38:	889		993		998	998	998			998		998			
39:	.00	.00	.25	.25	.50										
40:	.00	.00	.25	.25	.50										
41:	.00	.00	.25	.25	.50										
42:	.00	.00	.25	.25	.50										
43:	.00	.00	.25	.25	.50										
44:	.00	.00	.25	.25	.50										
45:	.00	.00	.25	.25	.50										
46:	A24044	AIR	COND	14000	BTU					01833	30	0	1	2	6
47:	53		53		53	55	55			55		55			
48:	.00	.00	.00	.50	.50										
49:	.00	.00	.00	.50	.50										
50:	.00	.00	.00	.50	.50										
51:	.00	.00	.00	.50	.50										
52:	.00	.00	.00	.30	.70										
53:	.00	.00	.00	.30	.70										
54:	.00	.00	.00	.30	.70										
55:	A24318	AIR	COND	18000	BTU					01833	30	0	1	2	7
56:	15		25		42	57	63			83		97			
57:	.00	.00	.25	.75	.00										

Figure III.1.5

UNCLASSIFIED\*\*\*EXAMPLE OF THE OUTPUT DATA FROM UTILITY CONTROL/COMPILER\*\*\*UNCLAS

```

1:TEMPORARY CONTROL DATA FILE WARP P88112DEC
2:  7   6   9  10   0   2  -1
3: 15  15  30  30  30  30  30
4:  1   1
5:  2   2
6:  1   2
7:  3   3
8:  4   4
9:  5   5
10:  6   6
11:  7   7
12:  1   4
13:  5   7
14: .05 .05 .05 .01 .01 .00 .00
15: .15 .15 .23 .10 .05 .02 .00
16: .15 .15 .10 .10 .05 .05 .05
17: 48.1 51.5 52.0 52.0 52.0 52.0 52.0
18: .000 .141 .059 .800
19: .255 .145 .050 .600
20: .200 .155 .005 .640
21: .100 .055 .105 .840
22: .005 .055 .140 .800
23: .305 .005 .000 .690
24: .555 .000 .000 .445
25: .000 .994 .211 1.666 .222 .111 .666 .090 .138 .899 .574
26: .224 .691 .254 .714 .999 .744 .359 .807 .644 .921 .100
27: .100 .992 .600 2.744 .578 .000 .299 .000 .704 .000 2.192
28: 7.682 3.339 .668 1.750 1.199 .502 .078 .515 .597 .000 .080
29: .200 2.507 1.254 .927 .494 .000 .633 .000 .283 .284 1.609
30: 6.753 3.355 .895 3.001 1.679 1.789 .465 .899 .566 .000 .090
31: .050 .255 .064 .555 .888 .010 .088 .550 .923 .760 .064
32: 3.142 7.460 .533 .850 .973 .349 .077 .050 .778 .000 .080
33: .649 .764 .980 1.000 .950 .485 .196
34: 64.00 1.40 .00 .00 10.00 24.00 36.00 72.00 7.90 .00 9.00 .00 .00
35: 56.00 1.40 .00 2.00 20.00 34.00 52.00 60.00 4.20 .00 8.00 .00 .00
36: 50.00 .90 .00 2.00 15.00 24.00 37.00 28.00 2.30 .00 7.00 .00 .00
37: 27.00 .50 .00 .00 6.00 18.00 25.00 10.00 .60 .00 .00 .00 .00
38: 19.00 .40 .00 .00 5.00 13.00 16.00 4.00 .50 .00 6.00 .00 .00
39: 30.00 .30 .00 4.00 4.00 14.00 33.00 3.00 .10 .00 4.00 .00 .00
40: 17.00 .10 .00 2.00 3.00 7.00 15.00 1.00 .10 .00 1.00 .00 .00

```

Figure III.1.6

UNCLASSIFIED\*\*EXAMPLE OF RATES-XX/SFC-1 OUTPUT DATA FROM UTILITY ELCON/3

```

1: TEMPORARY CONTROL DATA FILE WARP P08112DEC
2:
3:
4:
5:
6: NPER MXITH 0 CEM CLASSES WARP SETS
7: 7 6 9 10
8:
9:
10: DAYS PER PERIOD
11: 15 15 30 30 30 30 30
12:
13:
14: FIRST AND LAST PERIOD FOR EACH WARP SET
15: 1 1
16: 2 2
17: 1 2
18: 3 3
19: 4 4
20: 5 5
21: 6 6
22: 7 7
23: 1 4
24: 5 7
25:
26:
27: AIR LOSS RATE BY PERIOD
28: .050 .050 .050 .010 .010 .000 .000
29:
30:
31: SEA LOSS RATE BY PERIOD
32: .150 .150 .230 .100 .050 .020 .000
33:
34:
35: LOC LOSS RATE BY PERIOD
36: .150 .150 .100 .100 .050 .050 .050
37:
38:
39: AVERAGE LARGE UNITS BY PERIOD
40: 48.10 51.50 52.00 52.00 52.00 52.00 52.00
41:
42:
43: FRACTION OF FORCE BY POSTURE BY PERIOD
44: ATTACK DEFEND WITHDRAW INACTIVE
45: .0000 .1410 .0590 .8000
46: .2550 .1450 .0500 .6000
47: .2000 .1550 .0050 .6400
48: .1000 .0550 .1050 .8400
49: .0050 .0550 .1400 .8000
50: .3050 .0050 .0000 .6900
51: .5550 .0000 .0000 .4450
52:
53:
54: ARTY KILLS (% PER DAY) BY VULN CLASS BY POSTURE
55:
56: .000 .994 .211 1.666 .222 .111 .666 .090 .138 .699 .574
57: .224 .691 .264 .714 .999 .744 .359 .807 .644 .921 .100
58:
59: .100 .992 .600 2.744 .578 .000 .299 .000 .704 .000 2.192
60: 7.682 3.339 .668 1.750 1.199 .502 .078 .515 .597 .000 .000
61:
62: .200 2.007 1.254 .927 .494 .000 .633 .000 .283 .284 1.609
63: 6.753 3.355 .895 3.001 1.678 1.789 .465 .898 .566 .000 .090
64:
65: .050 .255 .064 .555 .888 .010 .088 .550 .823 .760 .064
66: 3.142 7.460 .533 .850 .973 .349 .077 .050 .778 .000 .060
67:
68:
69:
70: ARTY SCALING FACTORS BY PERIOD
71: .649 .764 .980 1.000 .950 .485 .196
72:
73:
74: CEM KILLS (% PER 30 DAYS) BY CLASS BY PERIOD
75:
76: 64.000 1.400 .000 .000 10.000 24.000 36.000 72.000 7.800
77:
78: 56.000 1.400 .000 2.000 20.000 34.000 52.000 60.000 4.200
79:
80: 50.000 .900 .000 2.000 15.000 24.000 37.000 28.000 2.300
81:
82: 27.000 .500 .000 .000 6.000 18.000 25.000 10.000 .600
83:
84: 19.000 .400 .000 .000 5.000 13.000 16.000 4.000 .500
85:
86: 30.000 .300 .000 4.000 4.000 14.000 33.000 3.000 .100
87:
88: 17.000 .100 .000 2.000 3.000 7.000 15.000 1.000 .100

```

Figure III.1.7

UNCLASSIFIED\*\*\*EXAMPLE OF RATES-XX/MONTHLY-WOL OUTPUT DATA FROM UTILITY ELCON/3

1:	A07198 AK VEH M218 GH EG P1A	2.43	7.21	2.85	2.48	3.14	3.11	2.28	3.03	2.85	2.80
2:											
3:	A14752 ADAP TEST CAMER: LM178	8.04	10.65	9.64	9.79	10.44	10.26	9.08	7.06	10.00	8.47
4:											
5:	A27496 AIMING CIRCLE M2 W/E	30.73	40.37	35.58	39.79	49.34	46.76	21.56	18.44	42.10	28.86
6:											
7:	A23770 AIR COND FL/WNOM 6000B	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
8:											
9:	A27828 AIR COND F/WA 9000 BTU	1.15	1.77	1.48	1.44	2.14	2.10	.95	.87	1.69	1.31
10:											
11:	A24044 AIR COND 18000 BTU	.60	.74	.67	.47	.83	.74	.33	.47	.66	.51
12:											

Figure III.1.8

UNCLASSIFIED\*\*\*EXAMPLE OF THE RATE S-XX/MONTHLY-WL OUTPUT DATA FROM ELCON/3

1: A03198 AK VEH M218 GM EQ P1A	2.89	4.52	5.94	3.27	11.82	6.57	3.15	3.31	7.18	4.21
2:										
3: A14752 ADAP TEST CAMERA LM178	9.70	24.23	18.63	13.13	17.44	12.51	9.14	7.84	16.54	9.77
4:										
5: A22496 AIMING CIRCLE M2 W/E	37.08	49.58	43.34	57.60	73.84	56.98	24.38	20.51	59.07	33.28
6:										
7: A23770 AIR COND FLYWIND 6000 B	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
8:										
9: A23828 AIR COND F/WA 9000 BTU	1.36	5.25	3.41	1.98	3.13	2.49	1.06	.94	2.87	1.47
10:										
11: A24044 AIR COND 18000 BTU	.71	.87	.79	.62	2.31	.89	.37	.50	1.21	.58
12:										

Figure III.1.9

THIS PAGE IS INTENTIONALLY LEFT BLANK

## Chapter 2

### WARF INTERMEDIATE MATERIEL PROCESSOR

**2.1 DESCRIPTION OF PROCESSING:** The WARF Intermediate Materiel Processor (WIMP) is the newest portion of the WARRAMP methodology, having been created in 1980 to supplant the Red Artillery Model (RAM) Methodology. This program's implementation is tied to the development of, and use of, full force array data (combat units, including support type units) in the COSAGE model. Modifications of this utility is anticipated as the further development of COSAGE occurs: for example, the addition of tactical air support modules to COSAGE. The source code employed in the WIMP is extractions from the COSAGE model. Hence the programmer will encounter some items (code) that has no direct bearing on the WIMP methodology, but are necessary for the completeness of the program.

**2.1.1 PURPOSE/FUNCTIONS:** The purpose of this program is to process the red force's artillery (indirect fire) missions (a special output from COSAGE) on the blue force array (unit data) with the equipment of the units redefined as one of the (22 each) artillery vulnerability groups, which yields (as an output) a re-stated (by vulnerability group) set of unit losses.

COSAGE treats the number of equipment types as a variable and each notional equipment type belongs to a generic family possessing similar characteristics for direct fire and indirect fire vulnerabilities. However, the WARF methodology treats equipment types as a fixed set (22 each) of specific groups according to artillery (indirect) fire vulnerability. Also, the units modeled in COSAGE have only combat equipment (major) items and it is necessary to estimate losses of other essential, but non-combat (non-firepower producing) items of equipment. For these reasons, a units table of organization and equipment (TOE) is restated from the COSAGE methodology and put into the WARF methodology context.

The output (unit losses by vulnerability group) is then processed by the program WIMP/LOSS-RATES to yield the average equipment loss and a rate of loss over time (period of ground combat). Though the WIMP processor program is implemented in the SIMSCRIPT II.5 programming language, it does not employ the event or process (timing routines) features of SIMSCRIPT II.5. It performs the data input functions, processes the data computationally and writes the results (output) in a format ready for the next MPP processor. It is a one-sided program in the WARRAMP methodology, the only interest being the effects of the red force's artillery fire on the blue force.

**2.1.2 PROGRAM INPUT/OUTPUT STRUCTURE:** The WIMP I/O structure is depicted in Figure III.2.1. This general graphic presents the input/output file-name.element-name normally associated with the programs operation.

**2.1.2.A INPUT DATA AND DATA BASE:** The following input files are required. These files are generally free-formatted. This feature demands user discipline in maintaining a data organization for ease of editing and visual error checking. The variables associated with the input are defined in the Data Element Dictionary, paragraph 2.1.2.C. A summary of I/O is presented in table III.2.1.

- o Category, Type Unit (CATTU) Data. This file is a free-formatted data file with mixed mode data that provides the parameters for the unit-specific data that will follow in the .CAT\$TU data file. This data file is a copy of the COSAGE Data Base file by the name UNIT\$DATA. The categories and types of units have SIMSCRIPT II.5 defined attributes and set memberships by their definition as a permanent or temporary entity. These relationships are depicted in Figure III.2.2. A sample of the data is presented in Figure III.2.8. This data is read into the program by the routine CAT.TU.INPUT.
- o Equipment and Type Equipment (EQUIP) Data. This data file is a free formatted data file with mixed mode data values that provides the parameters and descriptors for the equipment items assigned to each unit in the .UNITXXX file and values necessary for the ground combat modeling performed in COSAGE. This data file is prepared by the WARF analyst to reflect the 22 equipment vulnerability categories and personnel. The types equipment and equipment items are SIMSCRIPT II.5 defined permanent entities with attributes and set relationships as depicted in figure III.2.2. A sample of the data file is presented in Figure III.2.9. This data is read into the program by the routine EQ.TE.-INPUT.
- o Unit Data (.UNITXXX). This file is a free formatted data file with all input values being integers that contain the force array data. The force is the grouping of units in an order-of-battle hierarchy with each unit having certain features (attributes). This data file is a copy of the COSAGE Data Base file by the name UNIT\$DATA. The element name has the "XXX" appended to enable user's to change the name by changing these characters to a label identifying the data file by a combat posture. A posture denoting the type of array by the modeled disposition of the forces, i.e., attack, delay, defense intense or defense light. The units in this file are sorted by the side identification because the analytical interest is in the blue force data. The units are SIMSCRIPT II.5 permanent entities, have attributes and set memberships as depicted in figure III.2.3. A sample of the data file is contained in figure III.2.10. This data is read into the program by the routine UNIT.INPUT.
- o Posture, Environment and Mission Data (PEM). This file is a free-formatted file that contains mixed mode data. This data is a copy of the COSAGE Data File by the name P\$E\$M\$DATA. The purpose of the data is to initialize the types of postures (standing, open, foxholes, in tanks, etc.) that will be modeled for personnel, and the environments that will be modeled (open, town, woods), and mission the tactical units may be directed to have (attack, delay, defend, withdraw, etc). These variables are used to establish arrays and then read in the particular data. In the SIMSCRIPT II.5 language, these are compound entities, or multidimensional arrays, representing distributions of a tactical units assets on the hypothetical battlefield. The organization is depicted in Figure III.2.4. A sample of the data file is in Figure III.2.X. This data is read into the program by the routine P.E.M.INPUT.

read into the program by the routine P.E.M.INPUT.

- o Type (Artillery - Indirect Fire) Battery Input Data (TBTRY). This file is a free-formatted file that contains mixed-mode data. This data is a copy of the COSAGE data base file by the name TB\$DATA. The purpose of this data is to initialize the types of indirect fire units that will be modeled among the blue and red force arrays. The SIMSCRIPT II.5 organization is depicted in Figure III.2.5. A sample of the data file is in Figure III.2.12. This data is read into the program by the routine TB-INPUT.
- o Battery (Artillery - Indirect Fire) Input (BTRYXX) Data. This file is a free-formatted file that contains mixed-mode data. It is a copy of the COSAGE data base file by the name BTRY\$DATA. The "XXX" portion of the file name is edited by the user to contain characters that relate the data file to a specific force array (posture) in the same manner that the UNITXXX file is named to provide a data audit trail. The purpose of the data file is to input the indirect firing units; the battery are so identified but are also one of the named units input in the UNITXXX data file. All of the modeled indirect firing units are input for both forces. The SIMSCRIPT II.5 relationships are depicted in Figure III.2.5 and the sample data file in Figure III.2.13. This data is read into the program by the routine BTRY.INPUT.
- o Munitions (for indirect firing systems) Input Data (MUNS). This file is a free-formatted file that contains mixed-mode data. It is a copy of the COSAGE data base file by the name MUN\$DATA. The purpose of this data is to input the data by munitions type for each type of (indirect firing) battery modeled. Each munition has attributes for range and lethality and has sets that contain the values for different fusing methods and error over range values. These relationships are depicted in Figures III.2.5 thru III.2.7, and the sample data file is in Figure III.2.14. This data is read into the program by the routine MUN\$INPUT.
- o Submunitions (SUBMUNS) Input Data. This file is a free - formatted data file that contains integer data. It is created by the WARF analyst using AMSAA provided catastrophic lethal area data. The purpose of the data file is to initialize the reliability and lethality of the modeled indirect fire submunitions. These submunitions relate only to the improved conventional munitions (ICM). The SIMSCRIPT II.5 relationships are depicted in Figure III.2.7. A sample data file is in figure III.2.15. This data is read into the program by the routine SUBM.INPUT.
- o High Explosive Lethal Area Input Data (HELA). This file is a free - formatted data file that contains integer data. It is created by the WARF analyst using AMSAA provided catastrophic lethal area data. The purpose of the data is to initialize the lethality data for the high explosive munitions modeled. The internal data structure relationships are depicted in Figure III.2.8. A copy of the sample input data is in Figure III.2.16. This data is read by the routine HE.LA.INPUT.

- o SIMU5-XX Input data contains the fire mission data that is the resultant output from a COSAGE simulation. This file is read as a free formatted data file with mixed-mode data. This data is the basis for the computations of the WIMP program, and it is read by the WARF.ARTY routine. A sample of this input data file is in Figure III.2.17.
- o WIMP/TOE-IN input data file contains the restated table of organization and equipment structure, according to the artillery vulnerability groups for each type unit modeled. This data is an output from the WIMP/TOE-IN program and a sample of this data is illustrated in Figure III.2.18. This data is read into the WIMP program by the WARF.ARTY routine.

**2.1.2.B OUTPUT DATA AND DATA FILES:** This program produces two output files from the program execution. They are as follows; refer to table III.2.1 for a summary.

- o Run output is produced from the demand mode - submitted run stream via the system PRINT\$ file; default is logical unit 6. Refer to Volume I, Figure III.2.2 for the runstream. Through the Executive 8 commands the user puts the WIMP program into execution and performs an edit of the successful program runs into a breakpointed file called "XXPRINT". The runstream itself must be edited on line by the user to substitute appropriate labels for output file element names in lieu of the "X" characters.
- o A print file cataloged as XXWIMPOUT (Figure III.2.33) is initialized by the program to capture the computed output of the WARF.ARTY routine. The output is written with formatting statements and all output data fields are decimal (real). Logical Unit 3 is applied for this print file; however the output data is edited (copied) into a file element version called .WIMP/XX-X at the conclusion of a normal program execution. The "X" labels on the version must be edited (in runstream) by the user and replaced by characters denoting the source (posture) UNITXXX input file and the run iteration number for an audit trail.

**WIMP**  
**INPUT/OUTPUT**

	LOGICAL UNIT	---FORMAT---		---DATA MODE---		
		FREE	SPEC	ALPHA	REAL	INTEGER
CATTU	5	X		X	X	X
EQUIP	5	X		X		X
UNITXXX	5	X				X
PEM	5	X		X		X
TBBTRY	5	X		X		X
BTRYXXX	5	X				X
MUNS	5	X		X		X
SUBMUNS	5	X		X		X
HELA	5	X				
WIMP/TOE-IN	5	X			X	
SIMO5-XX	4	X			X	X
XXPRINT	6		X	X	X	X
XXWIMPOUT	3		X		X	
WIMP/XX-X	6		X		X	

Table III.2.1

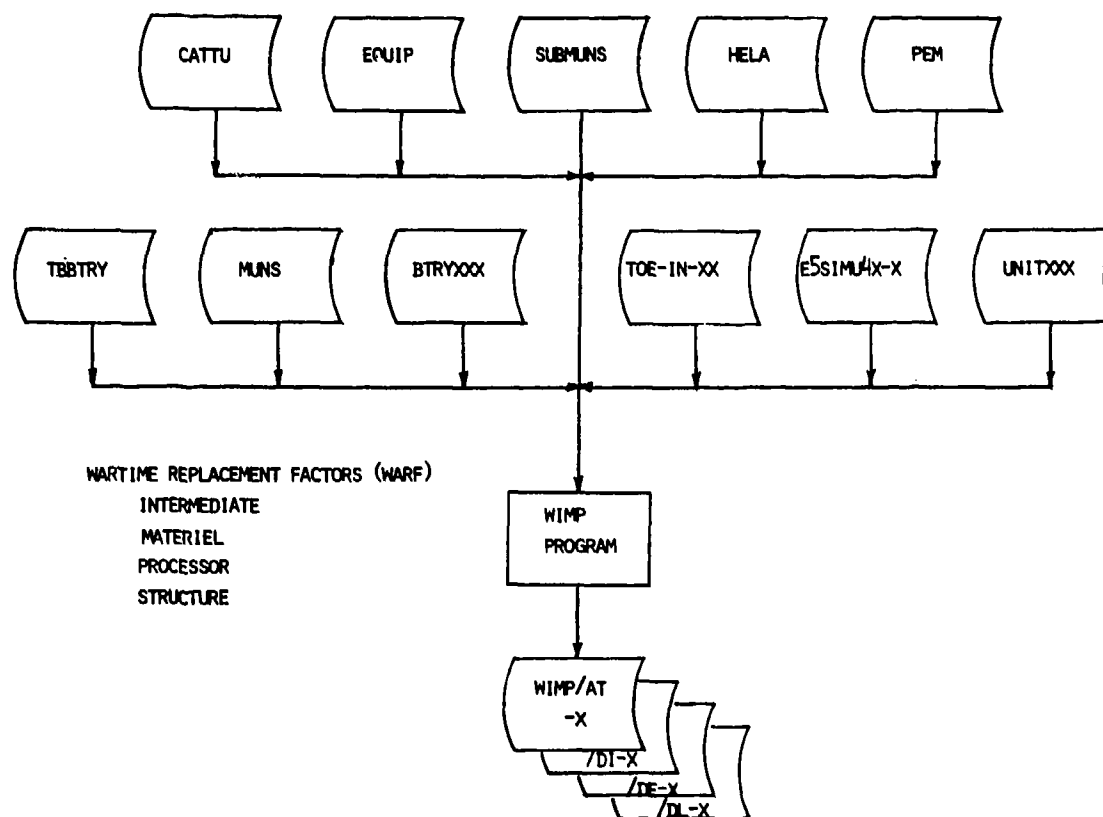


Figure III.2.1

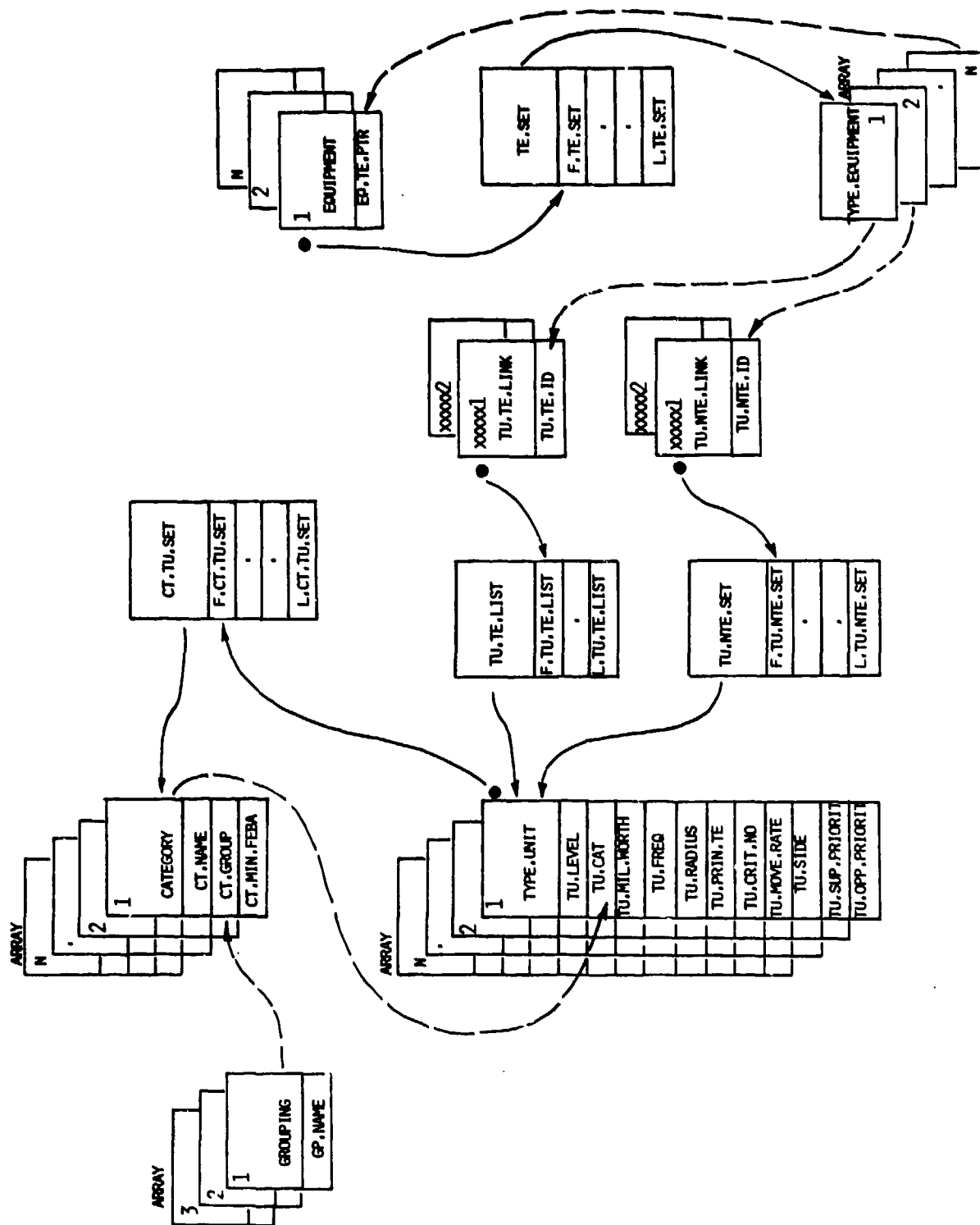


Figure III.2.2

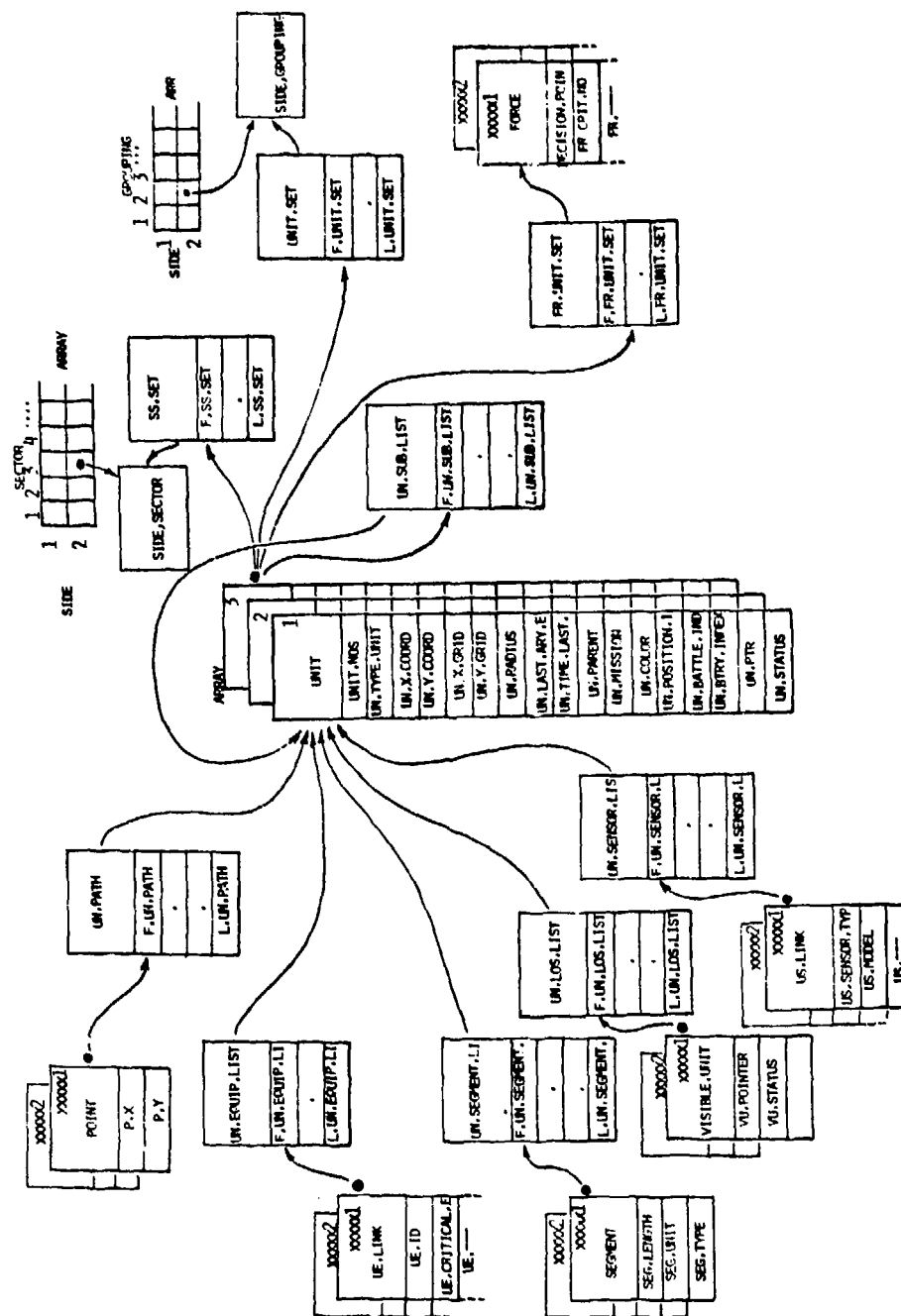


Figure III.2.3

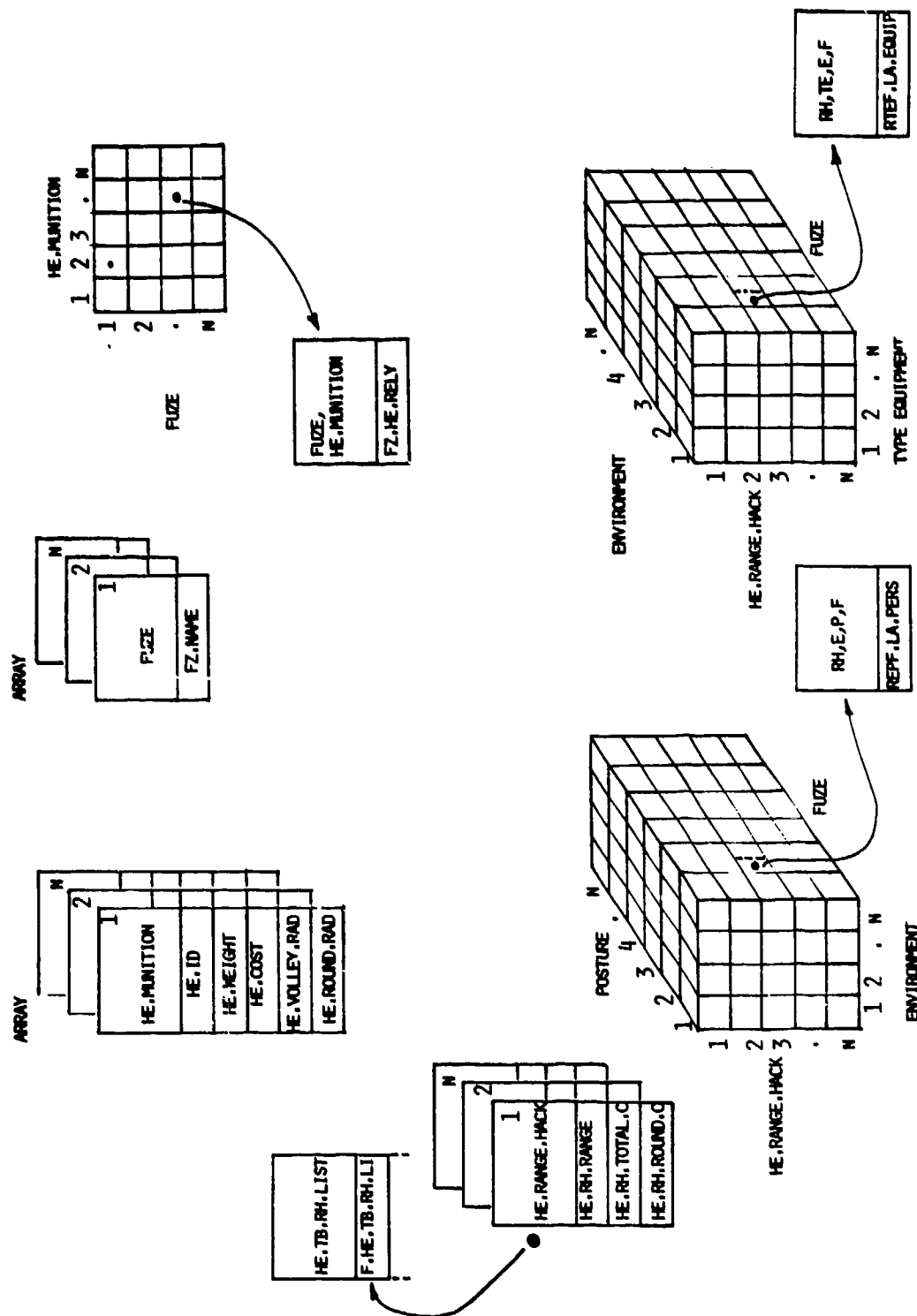


Figure III.2.4

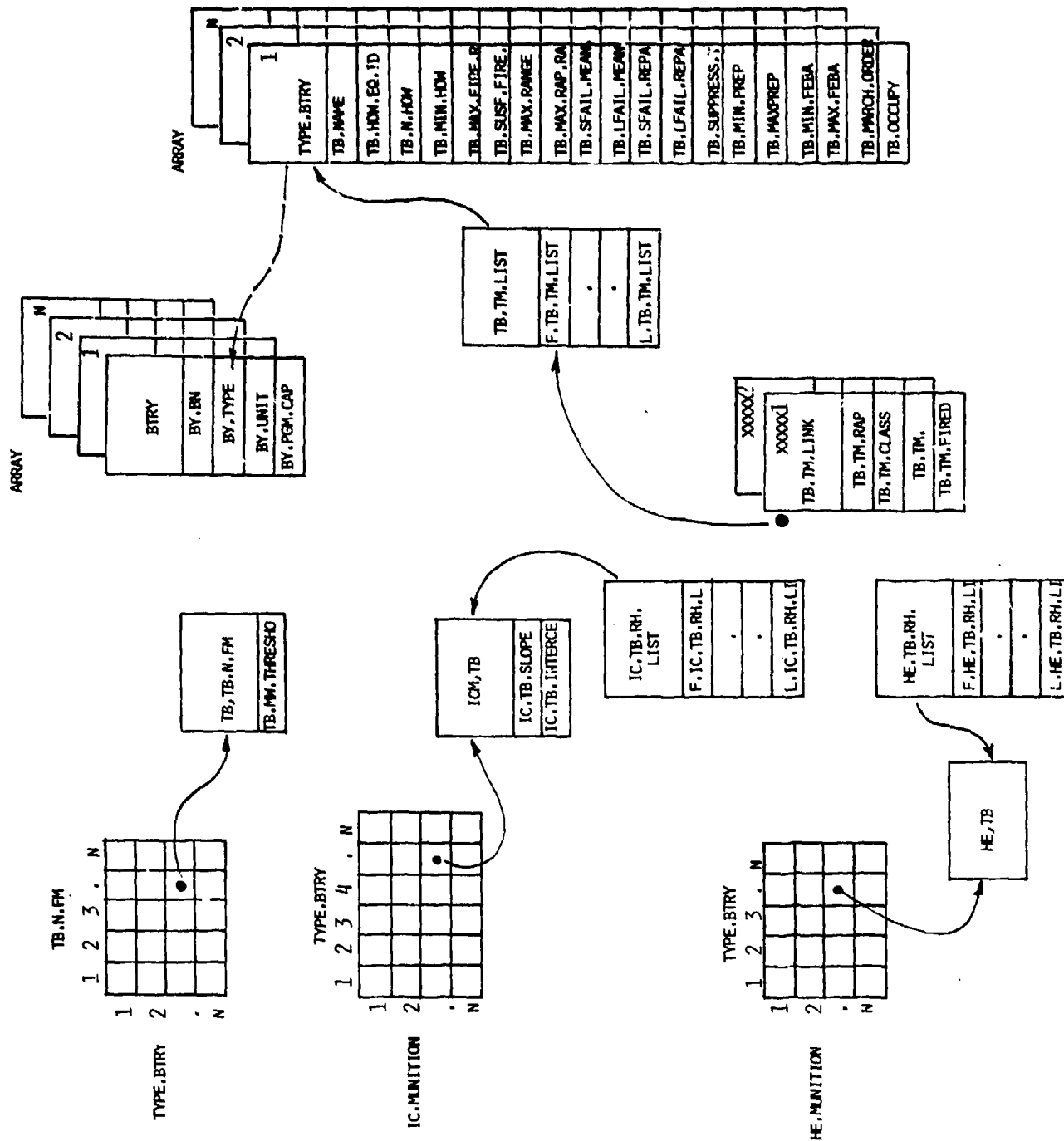


Figure III.2.5

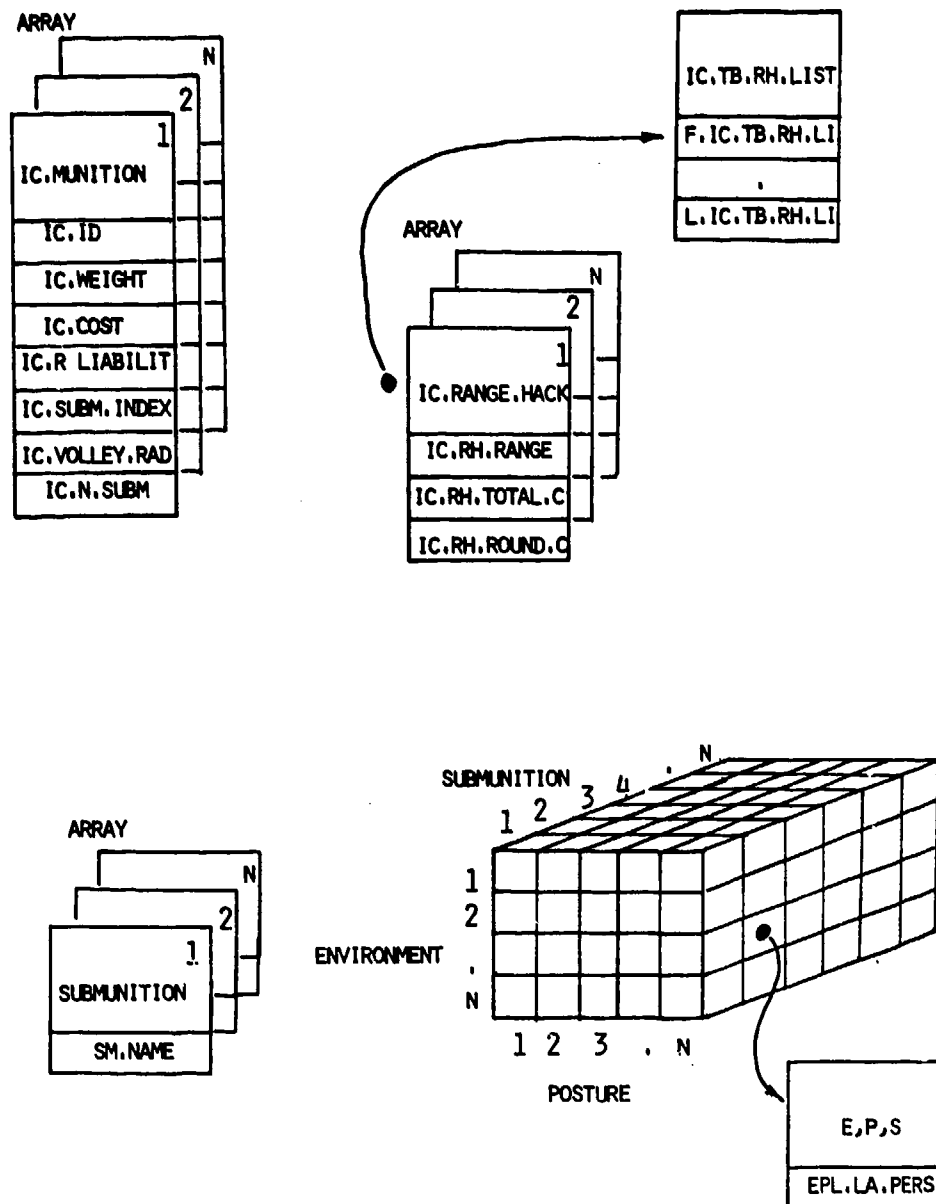


Figure III.2.6

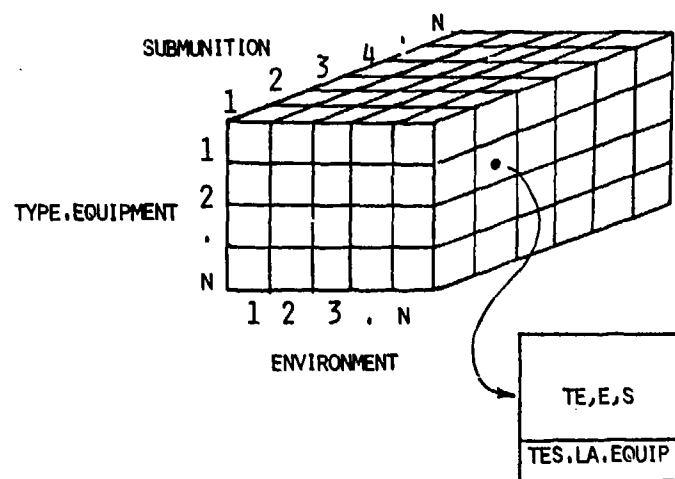
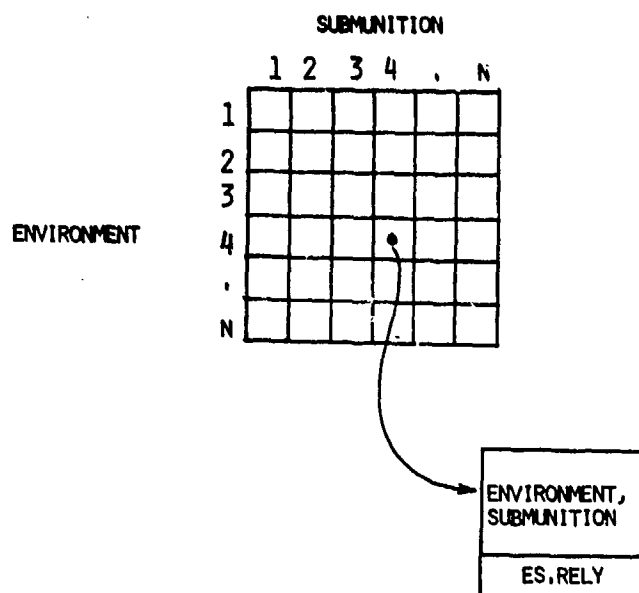


Figure III.2.7

**2.1.2.C DATA ELEMENT DICTIONARY:** The variables utilized with the WARF Intermediate Materiel Processor are defined in the SIMSCRIPT II.5 Preamble (Figure III.2.19). The Preamble established the data structure through the definition of permanent or temporary entities and attributes, and they may own and or be members of sets or lists of entities. Refer to the reference SIMSCRIPT II.5 manuals (ref S T, U) for details. The preamble declarations of the WIMP are extensive and beyond the requirements of the processor because it is, like the input routines, an adaptation from the COSAGE model. For this reason the data element dictionary will address only those items used specifically in the WIMP.

- o Global variables: The following variables are defined for use throughout (globally) the program.

<u>Descriptor word</u>	<u>Mode</u>	<u>Value</u>	<u>Definition</u>
ARTILLERY	Integer	2	One of the four categories of units.
BLUE	Integer	2	One of the two sides modeled.
BLUE.UNIT.CNTR	Signed Integer	VAR	A counter for the number of Blue Units.
CBTAVN	Integer	4	Abbreviation for "Combat Aviation" and one of the four categories of units.
DATA.ERROR	Signed Integer	Var (0,1)	A flag for input routines.
DEBUG	Signed Integer	Var (0,1)	A flag for selective writing of debugging information.
FALSE	integer	0	A switch value.
HIT	Integer	1	A discriptor for program text, a flag.
MANEUVER	Integer	1	One of the four categories of units; denotes those units that may move on the battlefield, infantry and Armor.

MISS	Integer	2	A descriptor for program text and a flag.
N.BTRY	Integer	0-N	The number of batteries (artillery) units modeled.
N.CATEGORY	Integer	0-N	The number of categories modeled.
N. ENVIRONMENT	Integer	0-N	The number of combat environments modeled.
N.EQUIPMENT	Integer	0-N	The number of equipment items modeled.
N.FUZE	Integer	0-N	The number of artillery HE munitions fuzes modeled.
N.GROUPING	Integer	3	The number of unit groups modeled; Artillery Manuever or Support.
N.HE.MUNITION	Integer	0-N	The number of high explosive munitions modeled.
N.HE.RANGE.HACK	Integer	0-N	The number of artillery HE range petitions modeled.
N.IC.MUNITIONS	Integer	0-N	The number of improved conventional (with submunitions) munitions modeled.
N.IC.RANGE.HACK	Integer	0-N	The number of artillery IC range partitions modeled.
N.MISSION	Integer	0-N	The number of missions (of units) modeled.

N.POSTURE	Integer	0-N	The number of personnel postures modeled.
N.SIDE	Integer	2	The number of forces or sides modeled.
N.SUBMUNITION	Integer	0-N	The number of submunitions for the ICM artillery rounds modeled.
N.TB.N.FM	Integer	5	The number of fire missions per type battery.
N.TYPE.BTRY	Integer	0-N	The quantity of types of indirect fire (artillery) batteries modeled.
N.TYPE.EQUIP	Integer	0-N	The quantity of types (groups) of equipment modeled.
N.TYPE.UNIT	Integer	0-N	The number of types of units modeled.
RED	Integer	1	One of the two sides modeled, the side color.
SUPPORT	Integer	3	One of the four groups of units.
TRUE	Integer	1	A flag, descriptor for the program text.
YES	Integer	1	A flag, descriptor for the program text.

- o Data Variables - Relationship of the following variables can easily be established by reviewing the appropriate Figure III.2.2 - III.2 - 7.

<u>Name</u>	<u>Mode</u>	<u>Definition</u>
BTRY	Integer - Permanent entity	An indirect fire unit; a pointer, value 1 to N.BTRY.
BY.BN	Integer - attribute	The number of the higher headquarters or the battalion to which a battery may belong.
BY.PGM.CAP	Integer - attribute	The precision guided munitions (laser guided) capability of the battery. May be "yes" or "no".
BY.TYPE	Integer - attribute	The type of battery value for this one battery; a value from 1 to TYPE.BTRY.
BY.UNIT	Integer - attribute	The value of the unit identification number of this unit from the unit data file. Identifies the battery to unit relationship.
CATEGORY	Integer - Permanent Entity	A definition of a group of units by their function; i.e. field artillery, mech infantry, a pointer.
CPM.UNWARNED	Integer - attribute	The percent of a unit according to category, posture and mission considered to be unwarned against an artillery attack. Percent times 100.

<u>Name</u>	<u>Mode</u>	<u>Definition</u>
CPM.WARNED.FRACT	Integer-attribute	The percent of a unit by category, posture and mission considered to be warned against an artillery attack. Percent times 100.
CT.GROUP	Integer-attribute	The grouping to which a category of units will belong; a pointer to the group. The defined value of the group 1, 2 or 3.
CT.MIN.FEBA	Integer-attribute	A distance, in decameters, from the forward edge of the battle area (FEBA) in which a category of units will be located.
CT.NAME	Alpha-attribute	The name, in 6 characters or less of the categories of units modeled.
EC.FRACT	Integer-attribute	The percent (times 100) of a unit by category, of equipment that may be found in a modeled environment.
ENVIRONMENT	Integer-Permanent Entity	One of the modeled environments that military units may be found in open, town or woods. Value from 1 to N.ENVIRONMENT.

EN.NAME	Alpha-attribute	The alphanumeric name (up to 6 characters) of a modeled environment.
EPS.LA.PERS	Integer-attribute	The lethal area, in square meters, times 10 of a submunition against personnel in a given environment and posture.
EQUIPMENT	Integer-Permanent Entity	One of the modeled equipments; value will be from 1 to N.EQUIPMENT.
EQ.TE.PTR	Integer-attribute	A pointer to the type of equipment (generic family) in which the equipment belongs. A value from 1 to N.TYPE.EQUIP.
ES.RELY	Integer-attribute	The probability times 100 of a submunition successfully detonating in a particular environment.
FUZE	Integer-Permanent Entity	One of the modeled fuses for indirect fire munitions, a value from 1 to N.FUZE.
FZ.HE.RELY	Integer-attribute	The probability times 100, of the fuze mounted on a high-explosive round, in denoting as prescribed.

<u>Name</u>	<u>mode</u>	<u>Definition</u>
FZ.NAME	Alpha-attribute	The alphanumeric name, up to 6 characters of the modeled indirect fire munitions fuses.
GP.NAME	Alpha-attribute	The alphanumeric name, up to 6 characters of a modeled unit grouping; Artillery, Maneuver Support.
GROUPING	Integer-Permanent Entity	A modeled group of military units according to their tactical function. Value from 1 to N.GROUPING. Refer to GP.NAME.
HC.WPN.TYPE	Integer-attribute	The pointer to the type of weapon(s) modeled on a type of helicopter. Value from 1 to N.TYPE.WEAPON.
HE.COST	Integer-attribute	The dollar cost of a high explosive munition. Used selectively in logic tests to select the least cost munition.
HE.ID	Alpha-attribute	The alphanumeric name, up to 6 characters given a high explosive munition.
HE.MUNITION	Integer-Permanent Entity	A modeled high explosive munition; value from 1 to N.HE.MUNITION.

<u>Name</u>	<u>Mode</u>	<u>Definition</u>
HE.RANGE.HACK	Integer-Permanent Entity	A modeled range partition for high explosive munitions based on the fact that munitions effective and firing system error is a function of range.
HE.RH.RANGE	Integer-attribute	A range, in decameters applicable to a high explosive munitions range back (partition).
HE.RH.ROUND.CPE	Integer-attribute	The circular probable error in delivery defined in meters, of a high single explosive round within a range back (partition).
HE.RH.TOTAL.CPE	Integer-attribute	The circular probable error in meters in the delivery of any quantity of high explosive rounds within a range back (partition).
HE.ROUND.RAD	Integer-attribute	The burst radius of a high explosive munitions - one round in meters.
HE.VOLLEY.RAD	Integer-attribute	The aggregate burst radius of a volley of high explosive minitions assuming the battery is deployed in a constant formation and firing a constant sheaf.

<u>Name</u>	<u>Mode</u>	<u>Definition</u>
HE.WEIGHT	Integer-attribute	The weight of the artillery round, in standard units - normally pounds. Used to compute the tonnage of munitions expended.
IC.COST	Integer-attribute	The dollar cost of a improved conventional munition.
IC.ID	Alpha-attribute	The alphanumeric description, up to 6 characters, of an improved conventional munition.
IC.MUNITION	Integer-Permanent Entity	A modeled improved conventional indirect fire munition; value from 1 to N.IC.MUNITION.
IC.N.SUBM	Integer-attribute	The quantity of submunitions contained within an improved conventional munition.
IC.RANGE.HACK	Integer-Permanent Entity	A modeled range partition for an improved conventional munition.
IC.RELIABILITY	Integer-attribute	The probability times 100 of an improved conventional munition detonating.

<u>Name</u>	<u>Mode</u>	<u>Definition</u>
IC.RH.RANGE	Integer-attribute	The range, in decameters of an improved conventional munition range partition.
IC.RH.ROUND.CPE	Integer-attribute	The circular error probable, in meters of an improved conventional munitions within a range back (partition).
IC.RH.TOTAL.CPTE	Integer attribute	The circular error probable, in meters, of an improved conventional munition volley within a range back (partition).
IC.SUBM.INDEX	Integer-attribute	A pointer to the submunition that is a component of the IC.MUNITION; value from 1 to N.SUBM.
IC.TB.INTERCEPT	Real-attribute	A value used to compute the effectiveness of the improved conventional munition; a function of the firing systems trajectory by type battery.
IC.TB.SLOPE	Real-attribute	The slope of the delivery trajectory of a type firing battery for improved conventional munitions.

<u>Name</u>	<u>Mode</u>	<u>Definition</u>
IC.VOLLEY.RAD	Integer-attribute	The radius of effects of an improved conventional munitions volley of rounds, in meters.
IC.WEIGHT	Integer-attribute	The firing weight of the improved conventional munition round, a standard measure, normally in pounds.
MAN.UNIT	Integer-Temporary Entity	A maneuver unit, modeled; one of the units input. Value is machine address.
MISSION	Integer-Permanent Entity	A modeled feature of military units; a value from 1 to N.MISSION.
MN.NAME	Alpha-attribute	The alphanumeric name, up to 6 characters of the mission.
MU.CRIT.NO	Integer-attribute	The quantity of critical or combat essential items belonging to a maneuver unit.
MU.REINF.IND	Integer-attribute	An index denoting that the maneuver unit can or cannot provide reinforcement to a unit requiring reinforcement.
POSTURE	Integer-Permanent Entity	A modeled feature of military units; a value from 1 to N.POSTURE.

<u>Name</u>	<u>Mode</u>	<u>Definition</u>
PT.NAME	Alpha-attribute	The alphanumeric name given to a modeled posture, up to 6 characters.
REPF.LA.PERS	Integer-attribute	The lethal area in square meters, times 10 of a high explosive munition given a fuze, range partition, posture and environment against personnel.
RTEF.LA.EQUIP	Integer-attribute	The lethal area in square meter times 10, of a high explosive munition, given a fuse, environment, and range back (partition) against a type of equipment.
SIDE	Integer-Permanent Entity	A modeled force, blue or red (2, 1).
SM.NAME	Alpha-attribute	A alphanumeric name, up to 6 characters of submunitions.
SUBMUNITION	Integer-Permanent Entity	A modeled feature of improved conventional munitions.
TB.HOW.EQ.ID	Integer-attribute	The type battery's pointer to the type of equipment that identifies its weapon system; a value from 1 to N.EQUIPMENT.

<u>Name</u>	<u>Mode</u>	<u>Definition</u>
TB.LFAIL.MEAN.RNDS	Integer-attribute	The long term, or high number of rounds that can be fired by a weapon system of the type battery, and a firing system failure will occur.
TB.LFAIL.REPAIR	Integer-attribute	The long term, or high hours (times 10) necessary to effect the repair of a weapon system of the type battery. High value of a uniform distribution.
TB.MARCH.ORDER	Integer-attribute	The time in minutes (a delay) necessary for the type of battery to be prepared to move or displace from one firing position to another.
TB.MAX.FEBA	Integer-attribute	The distance, in decameters that is the maximum allowable for the type battery to be located from the FEBA. A unit identified as this type battery, that exceeds this distance, will be issued an order to displace forward, or towards the FEBA.

<u>Name</u>	<u>Mode</u>	<u>Definition</u>
TB.MAX.FIRE.RATE	Integer-attribute	The maximum rate of fire for the type of battery, given a firing system. The highest sustained rate of fire for the total firing systems in rounds per minute.
TB.MAX.PREP	Integer-attribute	The maximum amount of time (minutes times 10) that a type of battery will require to prepare for a given fire mission. Upper value for a uniform distribution.
TB.MAX.RANGE	Integer-attribute	The maximum range that a firing system of the type battery, in decameters.
TB.MAX.RAP.RANGE	Integer-attribute	The maximum range that a firing system of the type battery can fire a rocket - assisted projectile, in decameters.
TB.MIN.FEBA	Integer-attribute	The distance, in decameters that is the minimum allowable for a battery of this type. A movement order will be generated to move such a battery a distance to the rear (array) from the FEBA.

<u>Name</u>	<u>Mode</u>	<u>Definition</u>
TB.MIN.HOW	Integer-attribute	The quantity of firing systems that is the minimum allowable for a battery of this type to execute fire orders. A battery that has less than this quantity is a non-firing unit. The losses may be permanent or temporary. Temporary implies that the howitzers are in repair.
TB.MIN.PREP	Integer-attribute	The minimum amount of time, in minutes (times 10) that a battery of this type will require to prepare for (a delay) a fire mission. The low end of a uniform distribution.
TB.MW.THRESHOLD	Integer-attribute of compound entities (2 - dim array)	A scaler value for this type of battery's military worth thresholds for the fire order queue. A method of ranking fire missions by the value of the target. A target having less than a minimum threshold value will not be fired upon by this type of battery.

<u>Name</u>	<u>Mode</u>	<u>Definition</u>
TB.NAME	Alpha-attribute	A user developed name for this type of battery, up to 6 characters. Normally the equipment type is associated with the name, e.g., 155HOW, etc.
TB.N.FM	Integer-permanent	The number of fire missions this type of battery may have. Set to a value of 5.
TB.N.HOW	Integer-attribute	The number of firing systems the type of battery will have at full (TOE) strength.
TB.OCCUPY	Integer-attribute	The amount of time, in minutes, (a delay) that a battery of this type will use in occupying a new firing position at the conclusion of a move.
TB.SFAIL.MEAN.RNDS	Integer-attribute	The mean number of rounds that a battery of this type will expend before a short term failure (minor) will occur.
TB.SFAIL.REPAIR	Integer-attribute	The mean time in hours (times 10) that a firing system of a type battery will be in repair for a short term (minor) failure. The total available firing systems of a battery of this type is decremented by one for the period.

<u>Name</u>	<u>Mode</u>	<u>Definition</u>
TB.TM.CLASS	Alphanumeric-attribute	The classification of type munitions for the type of batteries; up to six characters - the program expects, and uses HE or ICM.
TB.TM.LINK	Integer-temporary entity	A linkage created to relate type munitions to type batteries. The value is the true machine address.
TB.TM.RAP	Integer-attribute	A key value 0 or 1 to indicate that (1) the type munitions is a rocket assisted projectile.
TES.LA.EQUIP	Integer-attribute of a compound entity	The lethal area in square meters of a munition against a type equipment, in an environment, by a submunition of a ICM.
TU.CAT	Integer-attribute	The pointer to the category of units to which a type unit belongs. Value from 1 to N.Category.
TU.CRITICAL.EQUIP INDEX	Integer-attribute	A yes (1) or no (0) value set to determine if the type unit, type equipment link is a critical (combat) equipment item.
TU.CRIT.NO	Integer-attribute	A counter value containing the number of items of critical equipment that a type unit has.

<u>Name</u>	<u>Mode</u>	<u>Definition</u>
TU.FREQ	Integer-attribute	A counter value with the number of type units of this type modeled.
TU.LEVEL	Alphanumeric-attribute	A value, up to 6 characters that names the type of unit, e.g., CO, PLT, BNHO, etc.
TN.MIL.WORTH	Integer-attribute	A scalar value given to types of units by the user/analyst.
TU.MOVE.RATE	Integer-attribute	The maximum, unopposed rate in hexadecameters per hour; computation-ally used in unit movement, degraded by terrain, opposition and day or night factors.
TU.NTE.ID	Integer-attribute	The pointer to an equipment type that is not a type belonging to the type unit. Value of 1 to N.TYPE.EQUIPMENT.
TU.OPP.PRIORITY	Integer-attribute	A value based upon an analyst developed priority scheme for types of units.
TU.PRIN.TE	Integer-attribute	A pointer to the principle type of equipment employed by the type of unit; value of 1 to N.TYPE.EQUIPMENT.

<u>Name</u>	<u>Mode</u>	<u>Definition</u>
TU.RADIUS	Integer-attribute	The typical radius, in meters, of this type of unit in a combat deployment.
TU.SIDE	Integer-attribute	The pointer to the side to which this unit is represented within, value is 1 to 2.
TU.SUP.PRIORITY	Integer-attribute	A value based upon an analyst developed priority scheme for types of units in a support role.
TU.TE.ID	Integer-attribute	A pointer to the type of equipment the link represents; a value of 1 to N.TYPE.EQUIPMENT.
TU.TE.QUANT	Integer-attribute	The quantity of equipment of the type represented by the link.
TYPE.BTRY	Integer-entity	The pointer value of a type battery modeled; value from 1 to N.TYPE.BTRY.
TYPE.EQUIPMENT	Integer-entity	The pointer value of a modeled type of equipment; value from 1 to N.TYPE.EQUIP.
TYPE.UNIT	Integer-entity	The pointer value of a modeled type unit; value from 1 to N.TYPE.UNIT.
UNIT	Integer-entity	The pointer value of a modeled unit, value from 1 to N.UNIT.

<u>Name</u>	<u>Mode</u>	<u>Definition</u>
UN.LAST.ARTY.ENG	Integer-attribute	The time the unit was last engaged an opponent artillery unit; in minutes times 100.

2.1.3 PROGRAM PROCESSING: The high level program features are depicted in Figure III.2.32. The MAIN routine is the driver with the data definitions and background having been established by the PREAMBLE. With exception to the ERROR.STOP routine, all of the other routines perform the data input function. All computations are performed in the WARF.ARTY routine.

2.1.3.A PROGRAM RUN DESCRIPTION: The procedure or execution run file is discussed in Volume I of this documentation set, Chapter 2, of Section III. The file element resides in a permanently cataloged file and is designed to be executed in the demand mode from a computer terminal via the @ADD command. All data is read from logical unit 5 (READ\$) with exception to the E5SIMUX-X program file which reads data from logical unit 4. The computational output is temporarily captured in a permanent print file labeled XXWIMPOUT, in logical unit 3. It is subsequently edited (copied) to a file element called PF.WIMP/XX-X. The program run output is captured in a breakpointed file (logical unit 6 - PRINT\$), cataloged by the user as XXPRINT. The program calling and execution sequence is as depicted in Figure III.2.32. The program run output will be highlighted with processing statements that enable the user and programmer to follow the sequence of execution activity. These statements are:

- o DEBUG statements - The debug key is set to "TRUE" in the MAIN routine and all logical tests for printed debug statements will be satisfied. Thus the output will be written with these programmed statements with any specified variable on to unit 6.
- o PRINT statements - The routines of the program have over 22 unbounded print statements, directing that a formatted statement and/or variable be written onto the output file Unit 6.
- o LIST Statements - The SIMSCRIPT II.5 programming language enables programmers to incorporate "LIST" statements into the program. This statement will cause selected attributes of entities or an expression to be written onto the output unit 6 in a standard format with labels.
- o Bounded PRINT Statements - The routines have print statements bounded by programmed logic tests other than the Debug test. These logic tests, when satisfied will cause a programmer formatted statement and/or variable to be written on the output file, unit 6.

The program may stop prematurely by a subroutine call to the routine ERROR.-STOP by either the EQ.TE.INPUT routine if a logic test fails, or from the UNIT.-INPUT routine if a data logic test fails. A "STOP" may be executed in the routine BTRY.INPUT at two different locations if logic tests for data fails. If this occurs,

a normal job finish or termination is made.

**2.1.3.B PROGRAM LOGIC:** The program routines are coded with the structured IF-ELSE-ALWAYS (or synonyms) and controlled DO-LOOP's. A minimum number of GO TO 'LABELS' exist in the program (one in BTRY.INPUT). Some program routines have been coded with indentation to highlight the structure, all of which enables programmers to follow the logic as coded. In reviewing the language features coded, the maintenance programmer should review the substitutions created by the "Define to mean" statements in lines 614 through 632 of the PREAMBLE. The uses of syntax in the program differs from the SIMSCRIPT II.5 manuals because of these substitutions; all which make the program (text) logic more apparent. The variable background mode for all the routines and the preamble is declared as integer, which means that unless explicitly declared otherwise, all variables will be treated as an unsigned (but positive) integer value. Thus mixed mode computations or statements will yield positive integer values. The WIMP program is flow charted in Figures III.2.34 through III.2.45. In instances where expressions consist of a lengthy sequence of computations the word "COMP.VAL" appears. The programmer should refer to the source code for the expression.

**2.1.3.C PROCESSING FEATURES:** In addition to the features stated under the program run description, the following items are of note:

- o Variable packing: The variables named in the Preamble are not packed into fractions of words. This fact marginally (for this program) increases the amount of main core required, but also reduces the possibility of error.
- o Units of Measure:  
  
Ranges are expressed in 3 manners; 1) meters, the metric unit of measure, or 2) in decameters, which is number of meters divided by a factor of 10, or, 3) hexadecameters, which is defined as the number of decameters multiplied by 10 and divided by 16.  
  
Reliability and probabilities are expressed in integer value by multiplying the basic percent expression by 100 and rounding.
- o Data Bank: The dynamic storage allocation feature of SIMSCRIPT generally makes the DBANK information a matter of interest during a simulation; this program, which is computational, or accounting in nature, does not employ the timing routines. Thus the DBANK value is of marginal value. The DBANK is the number of words (decimal) rounded to the nearest thousand required by the data bank segments.
- o Programming conventions: The unusual variable name length capability of the SIMSCRIPT II.5 programming language enables the program to employ an english like text which facilitates understanding and comprehension. In general the first letters of an entity are used in the naming of its attributes; e.g., TYPE.BTRY has a TB.NAME and TYPE.UNIT has a TU.LEVEL. These naming conventions establish relationships useful in

reading and maintaining the program. Refer to the referenced SIMSCRIPT manuals for name length requirements.

**2.2 OPERATING ENVIRONMENT:** The program is executed via runstream submission in a demand mode environment from the computer terminal. Refer to Volume I for the runstream. At present the demand for main core memory is sufficiently low to permit this procedure and be within the operating policy. The system manages the program execution, and the runstream referred to in Volume I, contains all the necessary controls.

**2.2.1 HARDWARE:** The program executes on the UNIVAC 1100/82 OS with the fixed and removable disk storage space as requested in the runstream.

**2.2.2 SUPPORT SOFTWARE:** The program compilation and execution requires the following system processors:

@MAP	--	calls the MAP processor or collector to collect the relocatable object code.
@ED	--	call the editor processor
@SIM25	--	The SIMSCRIPT II.5 language processor
SDDL	--	The program may be processed through the Software Design and Documentation Language (SDDL) processor.

**2.2.2.A OPERATING SYSTEM:** The program was developed on, and tested on the UNIVAC 1100/82 OS. The standard features of the system (control language, processors, terminals, on-line storage) provide the essential support.

**2.2.2.B COMPILER:** The SIMSCRIPT II.5 is maintained on the USACAA system. This program was developed under the Release 6.3 compiler. Release 7.0 was installed in July 1981 and this program is compatible without modification, i.e., there are no compiler enhancements that are in conflict with the program.

**2.2.3 DATA BASE:** The input data for program execution resides in an informal data file constructed by the user/analyst. There are no formal data base requirements established for input or output.

**2.3 MAINTENANCE PROCEDURES:** The size of the program does not dictate extensive life cycle management procedures.

**2.3.1 PROGRAMMING CONVENTIONS:** The conventions employed throughout are applications of the SIMSCRIPT II.5 programming language with the structured programming techniques. Additionally, the following items are of note:

- o Indentation of code: With exception to the routine name (first line) and the end routine (last line) statement, the programmer should indent to highlight the program structure.

- o Comments are preceded by the ' ' (double apostrophe) characters and any characters following this notation in a record are ignored by the compiler.
- o The "define to mean" statements in the Preamble introduce words that can be employed in the code.
- o The programmer's initials and the dates of any changes to the code are commented and preceded by the percent symbol (%) to facilitate cross-referencing done by the SDDL processor.

**2.3.2 VERIFICATION PROCEDURES:** Program verification is achieved through a review of the run output (PRINT\$). The numerous debug, print, and list statements facilitate tracking the sequence of program execution and the verification of execution. Verification is completed through the hand calculations of the input data through the use of the coded expressions and a comparison with the program output.

**2.3.3 ERROR CORRECTION PROCEDURES:** Program debugging is achieved through the following techniques:

- o Debug statements - Source code level debugging is best accomplished through the use of coded debug statements at the approximate point of a suspected error. Data can be evaluated in this manner as well.
- o TRACE - The SIMSCRIPT II.5 library contains a tracing routine that traces back all subroutine calls from the error location and prints a dump, in octal, of the recursive storage for each routine. A verification of variable values can quickly be accomplished using the trace information along with the compiler listing. Refer to reference items.
- o SNAP.R - The SIMSCRIPT II.5 library contains a snapshot routine. The programmer may develop a substitute routine, but the routine provides lists of all the preamble declared entities, events, processes, etc. An examination of these variable values aid in the verification of the program and location of errors.
- o Post Mortem Dumps - The programmer may introduce the (@PMD (Exec 8) command in the runstream just after the data commands, and, in so doing received a printed listing of the contents of the program and data that were in main core at the time of error. The listing is in octal and must be manually interpreted. Refer to Appendix A, item U.
- o Data Errors - The free format feature of the input data necessitates that some structure be employed to aid in the visual inspection of data. User/analyst discipline in maintaining a format or structure is necessary. Otherwise trial and error techniques must be employed.

**2.3.4 SPECIAL MAINTENANCE PROCEDURES:** There are no established special procedures, save those outlined in the above paragraphs.

# AMP C110

1:	4	2222	0	1			
2:		CBKBR	0	1			
3:		CCCC	150	2			
4:		DDDD	0	1			
5:		EEEE	50	3			
6:		FFFF	0	1			
7:		GGGG	250	3			
8:		HHHH	2500	4			
9:		IIII	50	1			
10:							
11:	71	1	CDIV22	7	625 2000	02 500 2	0 0
12:					1 206 1	2 017 1	999
13:					3 4 5 6 7 8		999
14:	2		CDL72	2	750 1500	02 400 2	0 0
15:					2 008 1	4 001 2	999
16:					1 3 5 6 7 8		999
17:	3		CBKBRN	2	500 2000	02 300 2	0 0
18:					1 003 2	2 004 1	999
19:					3 4 5 6 7 8		999
20:	4		CBKBJJ	2	400 2000	02 200 2	4 3
21:					1 003 2	2 002 1	999
22:					3 4 5 6 7 8		999
23:	5		CBKCPA	2	300 2000	02 150 2	9 6
24:					1 005 2	2 004 1	999
25:					3 4 5 6 7 8		999
26:	6		CCCCRN	2	500 2000	02 300 2	0 0
27:					1 003 2	2 004 1	999
28:					3 4 5 6 7 8		999
29:	7		CCCCJJ	2	400 2000	02 200 2	4 3
30:					1 003 2	2 002 1	999
31:					3 4 5 6 7 8		999
32:	6		CCCCPA	2	300 2000	02 150 2	9 6
33:					1 002 1	2 004 1	999
34:					3 4 5 6 7 8		999
35:	9		CCCCPA	4	300 2000	02 200 2	9 6
36:					2 005 1		999
37:					1 3 4 5 6 7 8		999
38:	10		CCCCJJ	4	400 2000	02 200 2	4 3
39:					2 001 1		999
40:					1 3 4 5 6 7 8		999
41:	11		CEE-RN	1	110 2000	02 300 2	0 0
42:					1 001 2	2 003 1	3 002 1
43:					4 5 6 7 8		999
44:	12		CA11JJ	1	500 2000	03 200 2	3 4
45:					1 003 2	2 003 1	3 002 1
46:					4 5 6 7 8		999
47:	13		CA11PA	1	400 2000	03 150 2	5 7
48:					3 004 1		999
49:					1 2 4 5 6 7 8		999
50:	14		CXA3JJ	1	500 2000	03 200 2	3 4
51:					1 003 2	2 003 1	3 002 1
52:					4 5 6 7 8		999
53:	15		CXA3PA	1	400 2000	03 150 2	5 7
54:					3 004 1		999
55:					1 2 4 5 6 7 8		999
56:	16		SPA8BB	2	150 2000	02 300 2	0 0
57:					2 010 1		999

Figure III.2.8

# WIMP EQUIPMENTS

1:	23	AAAAAA	0	01	BBBBBB	0	09	CCCCC	0	10	DDDDDD	1	12	EEEEEE	0	08
2:		FFFFFF	1	10	GGGGGG	0	08	HHHHHH	0	03	IIIIII	0	10	JJJJJJ	0	05
3:		KKKKKK	1	10	LLLLLL	1	06	MMMMMM	0	05	NNNNNN	0	05	OOOOOO	0	01
4:		PPPPPP	0	05	QQQQQQ	0	02	RRRRRR	0	02	SSSSSS	0	05	TTTTTT	0	02
5:		UUUUUU	0	01	VVVVVV	0	10	WWWWWW	0	05						
6:																
7:	23 00	01	AAAAAA	01	0010	01	22	000								
8:		02	BBBBBB	02	1730	02	22	000								
9:		03	CCCCC	03	0200	11	22	000								
10:		04	DDDDDD	04	0200	04	22	000								
11:		05	EEEEEE	05	0250	02	22	000								
12:		06	FFFFFF	06	0250	02	22	000								
13:		07	GGGGGG	07	0100	03	22	000								
14:		08	HHHHHH	08	0200	00	22	000								
15:		09	IIIIII	09	0200	05	22	000								
16:		10	JJJJJJ	10	0100	00	22	000								
17:		11	KKKKKK	11	0200	02	22	000								
18:		12	LLLLLL	12	0200	02	22	000								
19:		13	MMMMMM	13	0250	03	22	000								
20:		14	NNNNNN	14	0250	02	22	000								
21:		15	OOOOOO	15	0010	01	22	000								
22:		16	PPPPPP	16	0200	05	22	000								
23:		17	QQQQQQ	17	0200	01	22	000								
24:		18	RRRRRR	18	0200	01	22	000								
25:		19	SSSSSS	19	0000	01	22	000								
26:		20	TTTTTT	20	0010	22	01	000								
27:		21	UUUUUU	21	0010	01	22	000								
28:		22	VVVVVV	22	0000	02	22	000								
29:		23	WWWWWW	23	0200	01	22	000								

Figure III.2.9

[illegible]

86

PEM

1:	3	AAAAA	BBBBB	CCCCC				
2:	3	DDDDD	EEEE	FFFF				
3:	6	GGGGG	HHHHH	IIIII	JJJJJ	KKKKK	LLLLL	
4:								
5:		20	80	73	20	90	80	20
6:		20	13	85	20	75	15	60
7:		73	05	05	75	00	05	20
8:								
9:		000	020	020	073	020	040	
10:		000	073	045	040	055	020	
11:		000	020	020	073	020	040	
12:		000	020	040	030	060	020	
13:		000	020	040	030	060	020	
14:		000	000	000	000	730	750	
15:								
16:		000	020	020	073	020	040	
17:		000	073	045	040	055	020	
18:		000	013	015	005	085	080	
19:		000	020	020	073	020	040	
20:		000	020	040	030	060	020	
21:		000	000	000	000	730	750	
22:								
23:		000	020	030	020	020	073	
24:		000	020	030	020	020	073	
25:		000	020	030	020	020	073	
26:		000	020	030	020	020	073	
27:		000	020	030	020	020	073	
28:		000	020	030	020	020	073	
29:								
30:								
31:		000	020	020	040	020	040	
32:		000	073	045	040	055	020	
33:		000	013	015	005	085	080	
34:		000	020	020	073	020	040	
35:		000	020	040	030	060	020	
36:		000	000	000	000	730	750	
37:								
38:		000	020	020	030	020	020	
39:		000	020	020	030	020	020	
40:		000	020	020	030	020	020	
41:		000	020	020	030	020	020	
42:		000	020	020	030	020	020	
43:		000	020	020	030	020	020	
44:								
45:		000	020	020	073	020	040	
46:		000	073	045	040	055	020	
47:		000	013	015	005	085	080	
48:		000	020	020	073	020	040	
49:		000	020	040	030	060	020	
50:		000	000	000	000	730	750	
51:								
52:		075	090	090	020	020	075	
53:		075	090	090	020	020	075	
54:		075	090	090	020	020	075	
55:		075	090	090	020	020	075	
56:		075	090	090	020	020	075	
57:		075	090	090	020	020	075	

Figure III.2.11

# LIMP TBTRY

```

1:23 1 WWWWWW 606021 1 1 60 30 3000 3000 200 500 2 35 05 50 150 600 700 1 1
2:      9 9 9 2002 2002 2002 2002 2002
3: 2 9992PD 606023 4 2 30 100 1110 2400 400 800 2 35 05 20 50 300 9990 5 1
4:      1 HE 07
5:      2 HE 01
6:      1 HE 01 9 9 9 000 005 2002 2002 2002
7: 3 999WHH 306021 6 1 60 300 1150 1150 200 500 2 35 05 20 50 200 700 10 5
8:      2 HE 01 9 9 9 000 005 2002 2002 2002
9: 4 999HHH 606022 9 2 30 100 2400 3000 300 500 2 35 05 20 50 300 9990 10 3
10:      1 HE 07
11:      2 HE 01
12:      1 HE 01 9 9 9 000 005 010 2002 2002
13: 5 11-MTC CCC22 2 1 100 800 0460 0460 100 500 2 35 05 05 10 050 0200 5 5
14:      2 HE 07 9 9 9 000 005 2002 2002 2002
15: 6 111LLL 606024 9 2 10 050 2070 3000 450 850 2 35 15 20 50 910 925 5 2
16:      2 HE 04
17:      2 HE 01
18:      1 HE 01 9 9 9 000 010 010 2002 2002
19: 7 AAACCC CCC21 6 1 100 800 0200 0200 100 500 2 35 05 05 10 050 0100 5 5
20:      2 HE 06 9 9 9 000 000 2002 2002 2002
21: 8 11-CCC CCC22 7 1 100 800 0460 0460 100 500 2 35 05 05 10 050 200 5 5
22:      2 HE 07 9 9 9 000 000 2002 2002 2002
23: 9 12-CCC CCC23 2 1 100 300 0565 0565 100 500 2 35 05 08 15 100 370 5 5
24:      2 HE 08 9 9 9 000 000 2002 2002 2002
25: 10 KKKZLR 606025 3 1 1 01 3500 3500 2000 5000 9 50 05 10 20 950 1000 2 2
26:      1 HE 01 9 9 9 000 005 010 2002 2002
27:
28: 11 999ZPH 606012 4 2 40 300 1530 1530 400 800 2 35 05 20 50 300 800 5 1
29:      2 HE 9 9 9 9 000 005 2002 2002 2002
30: 12 999THW 606011 2 2 40 300 1530 1530 200 500 2 35 05 10 20 300 800 10 3
31:      2 HE 9 9 9 9 000 005 2002 2002 2002
32: 13 888ZPH 606010 6 2 35 200 1850 1850 400 800 2 35 05 20 50 400 1500 5 2
33:      2 HE 01
34:      1 HE 01 9 9 9 000 005 010 2002 2002
35: 14 888THW 606015 9 2 35 200 1850 1850 400 800 2 35 05 10 20 400 1000 10 3
36:      2 HE 01
37:      1 HE 01 9 9 9 000 005 010 2002 2002
38: 15 999HOW 606017 6 7 05 060 2900 2900 400 800 2 35 05 20 50 1200 1800 5 5
39:      2 ICM 05
40:      1 ICM 11 9 9 9 000 005 010 2002 2002
41: 16 999ZPG 606014 5 3 55 300 2700 2700 400 800 2 35 05 20 50 1200 1600 5 2
42:      2 HE 10 9 9 9 000 005 010 2002 2002
43: 17 999PPP 606013 9 1 55 300 2700 2700 400 800 2 35 05 20 50 1100 1400 10 3
44:      2 HE 10 9 9 9 000 005 010 2002 2002
45: 18 888CCC CCC11 8 7 99 800 0304 0304 100 500 2 35 05 05 10 100 300 5 5
46:      2 HE 13 9 9 9 000 000 2002 2002 2002
47: 19 CCCCCC CCC12 5 9 99 100 0570 0570 100 500 2 35 05 05 10 100 300 5 5
48:      2 HE 14 9 9 9 000 000 2002 2002 2002
49: 20 CCCCCC CCC13 1 1 10 100 1000 1000 050 300 2 35 05 20 50 500 800 5 5
50:      1 HE 01
51:      1 HE 03 9 9 9 000 000 2002 2002 2002
52: 21 999III 888L11 7 2 50 03 2000 2000 1000 2000 3 40 05 10 20 550 600 5 5
53:      1 ICM 02 9 9 9 000 005 010 2002 2002
54: 22 999III 888L12 16 2 1 03 1000 1000 1000 2000 3 40 05 15 30 750 800 5 5
55:      1 HE 02
56:      2 HE 02 9 9 9 000 005 010 2002 2002
57: 23 999RRL 888L13 1 2 50 03 2000 2000 1000 2000 3 40 05 10 20 550 600 5 5
58:      1 ICM 02 9 9 9 000 005 010 2002 2002

```

Figure III.2.12



# WIMP MUNS

1:	17 7 2								
2:									
3:	XX YY								
4:	AA	042	174	090	090	98	90	0200	0048 0014 1150 0111 0111
5:	AA07	095	373	100	100	98	90	0400	0029 0015 2400 0050 0050
6:								0400	0029 0015 2400 0050 0050
7:	BB49	096	430	100	100	98	90	0800	0035 0021 3000 0100 0100
8:								0800	0035 0021 3000 0100 0100
9:	AA06	200	958	100	100	98	90	0400	0025 0015 2070 0153 0061
10:	CCC50	200	1081	100	100	98	90	0400	0025 0015 3000 0100 0100
11:	DDD20	007	276	050	040	98	90	0100	0025 0019 0200 0025 0019
12:	EE74	009	276	060	040	98	90	0200	0025 0019 0460 0030 0024
13:								0200	0025 0019 0460 0030 0024
14:	EEFFF	022	412	060	040	98	90	0400	0032 0030 0565 0047 0018
15:									
16:	GGGGGG	082	065	090	090	98	90	0200	0048 0014 1530 0120 0120
17:								0200	0048 0014 1530 0120 0120
18:	HHHHHH	090	070	090	090	98	90	0600	0063 0037 2700 0305 0178
19:								0600	0063 0037 2700 0305 0178
20:	IIIIII	020	095	100	100	98	90	0400	0029 0015 1850 0050 0050
21:								0400	0029 0015 1850 0050 0050
22:	JJJJJJ	300	210	100	100	98	90	0400	0025 0015 2900 0153 0061
23:	KKKKKK	009	041	060	040	98	90	0200	0025 0019 0304 0030 0024
24:	LLLLLL	022	075	060	040	98	90	0400	0032 0030 0570 0047 0018
25:	MMMMMM	80	200	100	100	98	90	0400	0032 0030 1000 0150 0060
26:	NNNNNN	020	200	100	100	95	85	0460	0083 0046 2000 0150 0100
27:								0460	0083 0046 2000 0150 0100
28:	000000	60	300	100	100	95	85	0800	0046 0030 1000 0100 0100
29:									
30:	AAAA	103	421	97	1	53	088	2.6	21. 400 038 011 2400 235 097
31:								2.6	21. 400 038 011 2400 235 097
32:	BR09	206	1027	97	2	53	195	3.3	32. 400 034 013 2070 144 039
33:	QQQQ	678	1282	97	3	200	1920	2.6	21. 500 105 091 3500 294 190
34:	IC152	1000	180	97	4	53	088	2.6	21. 400 038 011 2900 235 097
35:								2.6	21. 400 038 011 2900 235 097
36:	RRR03	900	180	97	4	53	195	3.3	32. 400 034 013 2070 144 039
37:	RRR40	900	170	97	4	53	195	3.3	32. 400 034 013 2070 144 039
38:	RRR20	1000	170	97	4	200	1200	2.6	21. 500 105 011 3500 294 190

Figure III.2.14

# WIMP SUBMUNS

1:	0	XXXX	YYYY	2222	XXXX		
2:							
3:		97	15	12	17	15	12
4:							
5:		9422	442	12	2762	342	14
6:		9422	442	12	2762	342	14
7:		9422	442	12	2762	342	14
8:		9422	142	12	2762	452	14
9:							
10:		942	112	212	422	422	422
11:		942	112	212	422	422	422
12:		942	112	212	422	422	422
13:		942					
14:							
15:		252	262	262	292	212	212
16:		252	262	262	292	212	212
17:		252	262	262	292	212	212
18:		252					
19:							
20:		272	262	262	292	212	212
21:		272	262	262	292	212	212
22:		272	262	262	292	212	212
23:		272					
24:							
25:							
26:		942	112	212	422	422	422
27:		942	112	212	422	422	422
28:		942	112	212	422	422	422
29:		942					
30:							
31:		252	262	262	212	212	212
32:		252	262	262	212	212	212
33:		252	262	262	212	212	212
34:		252					
35:							
36:		272	262	262	212	212	212
37:		272	262	262	212	212	212
38:		272	262	262	212	212	212
39:		272					
40:							
41:							
42:		142	112	212	422	422	422
43:		142	112	212	422	422	422
44:		142	112	212	422	422	422
45:		142					
46:							
47:		252	262	262	212	212	212
48:		252	262	262	212	212	212
49:		252	262	262	212	212	212
50:		252					
51:							
52:		272	262	262	212	212	212
53:		272	262	262	212	212	212
54:		272	262	262	212	212	212
55:		272					
56:							
57:							
58:		215	423	421	422	422	421
59:		422	422	421	423	142	212
60:		421	421	421	422	422	212
61:		422					

Figure III.2.15

# 41M1 HELA

1:	8361 0381	0240 0240	0282 0282
2:	8120 0120	0110 0110	0112 0112
3:	8010 0010	0002 0002	0009 0009
4:	8483 0483	0300 0300	0400 0400
5:	8396 0396	0200 0200	0220 0220
6:	8046 0046	0020 0020	0030 0030
7:			
8:	1268 1288	6628 0628	0642 0642
9:	8682 0682	0436 0436	0438 0438
10:	8101 0101	0060 0060	0021 0021
11:	1683 1683	8816 0816	0838 0838
12:	1126 1120	0266 0266	0268 0268
13:	8131 0131	0088 0088	0066 0066
14:			
15:	8609 0609	0299 0299	0302 0302
16:	8411 0411	0202 0202	0202 0202
17:	8048 0048	0022 0022	0048 0048
18:	8691 0891	0368 0368	0396 0396
19:	8234 0234	0262 0262	0266 0266
20:	8062 0062	0032 0032	0062 0062
21:			
22:	1846 1046	0602 0602	0223 0223
23:	8668 0668	0424 0424	0342 0342
24:	8082 0082	0023 0023	0083 0083
25:	1329 1329	8886 0886	0689 0689
26:	8893 0893	0290 0290	0448 0448
27:	8106 0106	0068 0068	0108 0108
28:			
29:	1826 1026	0422 0422	0213 0213
30:	8622 0622	0318 0318	0328 0328
31:	8081 0081	0039 0039	0081 0081
32:	1333 1333	8291 0291	0666 0666
33:	8621 0621	0413 0413	0426 0426
34:	8092 0092	0020 0020	0092 0092
35:			
36:	8621 0621	0361 0361	0411 0411
37:	8236 0236	0268 0268	0268 0268
38:	8028 0028	0031 0031	0028 0028
39:	1868 1068	0469 0469	0234 0234
40:	8696 0696	0348 0348	0348 0348
41:	8084 0084	0040 0040	0084 0084
42:			
43:	8614 0614	0498 0498	0220 0220
44:	8289 0289	0246 0246	0282 0282
45:	8004 0104	0102 0102	0103 0103
46:	8824 0824	0600 0600	0800 0800
47:	8429 0429	0300 0300	0382 0382
48:	8034 0034	0020 0020	0030 0030
49:			
50:	8263 0263	0418 0418	0200 0200
51:	8249 0249	0208 0208	0230 0230
52:	8008 0108	0102 0102	0106 0106
53:	8842 0842	0600 0600	0620 0620
54:	8469 0469	0300 0300	0320 0320
55:	8042 0042	0030 0030	0032 0032
56:			
57:			

Figure III.2.16

LIMP WIMP/A1-1

1:	2.39	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.
2:	2.	2.	36.22	2.	6.95	34.59	2.	4.22	2.	2.	2.	
3:												
4:	2.32	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.
5:	2.	2.	36.22	2.	6.94	34.57	2.	4.22	2.	2.	2.	
6:												
7:	36.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.
8:	2.	2.	36.	2.	7.	35.	2.	4.	2.	2.	2.	
9:												
10:	17.	2.	2.	2.	7.	3.	2.	3.	2.	2.	2.	2.
11:	2.	2.	45.	3.	7.	35.	2.	3.	2.	2.	2.	
12:												
13:	36.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.
14:	2.	2.	36.	2.	7.	35.	2.	4.	2.	2.	2.	
15:												
16:	5.43	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.
17:	2.	2.	36.22	2.	6.99	34.95	2.	4.22	2.	2.	2.	
18:												
19:	36.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.
20:	2.	2.	36.	2.	7.	35.	2.	4.	2.	2.	2.	
21:												
22:	17.	2.	2.	2.	7.	3.	2.	3.	2.	2.	2.	2.
23:	2.	2.	45.	3.	7.	35.	2.	3.	2.	2.	2.	
24:												
25:	36.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.
26:	2.	2.	36.	2.	7.	35.	2.	4.	2.	2.	2.	
27:												
28:	32.25	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.
29:	2.	2.	36.22	2.	6.99	34.99	2.	4.22	2.	2.	2.	
30:												
31:	2.37	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.
32:	2.	2.	36.22	2.	6.97	34.93	2.	4.22	2.	2.	2.	
33:												
34:	17.	2.	2.	2.	7.	3.	2.	3.	2.	2.	2.	2.
35:	2.	2.	45.	3.	7.	35.	2.	3.	2.	2.	2.	
36:												
37:	2.33	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.
38:	2.	2.	36.22	2.	6.93	34.55	2.	4.22	2.	2.	2.	
39:												
40:	36.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.
41:	2.	2.	36.	2.	7.	35.	2.	4.	2.	2.	2.	
42:												
43:	36.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.
44:	2.	2.	36.	2.	7.	35.	2.	4.	2.	2.	2.	
45:												
46:	17.	2.	2.	2.	7.	3.	2.	3.	2.	2.	2.	2.
47:	2.	2.	45.	3.	7.	35.	2.	3.	2.	2.	2.	
48:												
49:	33.47	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.
50:	2.	2.	36.57	32.79	23.55	25.49	3.97	26.97	2.	2.	2.	
51:												
52:	393.63	2.	6.99	2.	22.97	34.96	2.	25.22	2.	5.22	6.95	2.
53:	2.	33.93	273.22	34.95	69.96	323.54	37.22	33.22	3.22	2.	2.	
54:												
55:	36.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.	2.
56:	2.	2.	36.	2.	7.	35.	2.	4.	2.	2.	2.	
57:												

Figure III.2.17

## WIMP TOE/IN-AT

1:	09.99	.11	.30	.30	.30	.30	.30	.30
2:	.99	.11	.30	.30	.30	.30	09.30	.30
3:	7.99	15.11	.30	4.30	.30	.30	.30	
4:								
5:								
6:	09.99	.11	.30	.30	.30	.30	.30	.30
7:	.99	.11	.30	.30	.30	.30	09.30	.30
8:	7.99	15.11	.30	4.30	.30	.30	.30	
9:								
10:								
11:	09.99	.11	.30	.30	.30	.30	.30	.30
12:	.99	.11	.30	.30	.30	.30	09.30	.30
13:	7.99	15.11	.30	7.30	.30	.30	.30	
14:								
15:								
16:	76.99	.11	.30	.30	7.30	1.30	.30	1.30
17:	.99	.11	.30	.30	.30	.30	78.30	1.30
18:	7.99	38.11	.30	1.30	.30	.30	.30	
19:								
20:								
21:	09.99	.11	.30	.30	.30	.30	.30	.30
22:	.99	.11	.30	.30	.30	.30	09.30	.30
23:	7.99	15.11	.30	7.30	.30	.30	.30	
24:								
25:								
26:	09.99	.11	.30	.30	.30	.30	.30	.30
27:	.99	.11	.30	.30	.30	.30	09.30	.30
28:	7.99	15.11	.30	7.30	.30	.30	.30	
29:								
30:								
31:	09.99	.11	.30	.30	.30	.30	.30	.30
32:	.99	.11	.30	.30	.30	.30	09.30	.30
33:	7.99	15.11	.30	7.30	.30	.30	.30	
34:								
35:								
36:	76.99	.11	.30	.30	7.30	1.30	.30	1.30
37:	.99	.11	.30	.30	.30	.30	78.30	1.30
38:	7.99	38.11	.30	1.30	.30	.30	.30	
39:								
40:								
41:	09.99	.11	.30	.30	.30	.30	.30	.30
42:	.99	.11	.30	.30	.30	.30	09.30	.30
43:	7.99	15.11	.30	7.30	.30	.30	.30	
44:								
45:								
46:	09.99	.11	.30	.30	.30	.30	.30	.30
47:	.99	.11	.30	.30	.30	.30	09.30	.30
48:	7.99	15.11	.30	7.30	.30	.30	.30	
49:								
50:								
51:	09.99	.11	.30	.30	.30	.30	.30	.30
52:	.99	.11	.30	.30	.30	.30	09.30	.30
53:	7.99	15.11	.30	7.30	.30	.30	.30	
54:								
55:								
56:	76.99	.11	.30	.30	7.30	1.30	.30	1.30
57:	.99	.11	.30	.30	.30	.30	78.30	1.30

Figure III.2.18

LINE CACI SIMSCRIPT II.0 1100 SERIES RELEASE 6.3

```

1  PREAMBLE
2  ** WARTIME INTERMEDIATE PROCESSOR DATA STRUCTURES
3  ***** SECTION 1 ***** GENERAL *****
4  **
5  ** *****
6  ** * WARTIME REPLACEMENT FACTOR INTERMEDIATE PROCESSOR *
7  ** * PROGRAMMERS *
8  ** *
9  ** * MR ANDREW N. CARRAS MAR 60 - ANC *
10 ** * CPT SCOTT E. CANTON JAN 61 - SEC *
11 ** *
12 ** *****
13 **
14 **
15 ** PREAMBLE CONTENTS
16 **
17 ** SECTION II PERMANENT ENTITIES
18 ** SECTION III TEMPORARY ENTITIES
19 ** SECTION IV PROCESSES
20 ** SECTION V DEFINITIONS (GLOBAL)
21 ** A INTEGER VARIABLES
22 ** B SIGNED INTEGER VARIABLES
23 ** C ALPHA VARIABLES
24 ** D REAL VARIABLES
25 ** E SETS
26 ** SECTION VI SUBSTITUTIONS
27 ** SECTION VII COMPUTATION DIRECTIVES
28 **
29 **
30 ** ***** REMARKS *****
31 **
32 ** ***** THE WARTIME REPLACEMENT FACTOR INTERMEDIATE PROCESSOR IS
33 ** DESIGNED TO FUNCTION IN THE SYSTEM THAT COMPUTES WARTIME
34 ** REQUIREMENTS FOR AMMUNITION. IT COMPUTES LOSS RATES FOR
35 ** ITEMS NOT PLAYED IN THE CFM AND NOT CLASSIFIED AS ITEMS
36 ** WHOSE LOSS RATES ARE BASED ON HISTORICAL DATA.
37 ** THE FINAL ELEMENT IN THE COMPUTING SYSTEM IS THE EQUIP-
38 ** MENT LOSS CONSOLIDATOR (ELCON) WHICH IS DOCUMENTED
39 ** IN CAA-D-79-3 *****
40 LAST COLUMN IS 80 **FOR DATA INPUT FORMULAS
41 NORMALLY MODIF IS INTEGER
42
43
44 ***** SECTION 2 ***** PERMANENT ENTITIES*****
45
46
47 PERMANENT ENTITIES
48
49
50 EVERY RTTY HAS
51 A BY.BN,
52 A BY.TYPE,

```

Figure III.2.19

LINE CACI SIMSCRIPT IT-5 1100 SERIES RELEASE 6.3

```

53      A DY.UNIT,
54      A DY.PGM.CAP
55
56  EVERY CATEGORY HAS
57      A CT.NAME,          **ARMOR,MECH,S-ARTY,T-ARTY,ETC
58      A CT.GROUP,         **MANEUVER, ARTILLERY, SUPPORT
59      A CT.MIN.FEBA       **CLOSEST THIS TYPE UNIT COMES TO FEBA
60  OAINS
61      A CT.TU.SET
62  BELONGS TO
63      A GP.CAT.SET
64  HAS
65      A F.CT.TU.SET,
66      A L.CT.TU.SET,
67      A P.CT.TU.SET,
68      A P.GP.CAT.SET,
69      A S.GP.CAT.SET,
70      A M.GP.CAT.SET
71
72  EVERY CATEGORY , POSTURE , MISSION  HAS
73      A CPM.LARNED.FRACT,    **PERCENT OF CATEGORY IN POSTUP
74      A CPM.UNWARNED.FRACT  **PERCENT OF CATEGORY IN POSTURE WN
75
76  EVERY ENVIRONMENT HAS
77      A EN.NAME
78
79  EVERY ENVIRONMENT, CATEGORY  HAS
80      AN EC.FRACT
81
82  EVERY ENVIRONMENT,POSTURE,SUBMUNITION HAS
83      A EPS.LA.PERS          **LETHAL AREA AGAINST PERS-SOMXIO
84
85  EVERY ENVIRONMENT, SUBMUNITION HAS
86      A ES.RELY              **IN PERCENT
87
88  EVERY EQUIPMENT HAS
89      A EQ.TE.PTR
90  BELONGS TO
91      A TE.SET
92
93  EVERY FUZE HAS
94      A FZ.NAME
95
96  EVERY FUZE,HE,MUNITION HAS
97      A FZ.HC.RELY
98
99  EVERY HE,MUNITION HAS
100      A HE.ID,              **POUND DESIGNATION NO
101      A HE.WEIGHT,
102      A HE.COST,            **TO NEAREST $
103      A HE.VOLLEY.RAD,      **IN METERS
104      A HE.RC.IND.RAD       **IN METERS

```

Figure III.2.19 (cont)

```

LINE  CACI SINGSCRIPT II.F  1100 SERIES  RELEASE 6.3

105
106 EVERY HE.MUNITION,TYPE,RTY OWNS
107   A HE.TB.RH.LIST
108 HAS
109   A F.HE.TB.RH.LIST,
110   A L.HE.TB.RH.LIST,
111   A N.HE.TB.RH.LIST
112
113 EVERY HE.RANGE,HACK HAS
114   A HE.RH.RANGE,          **IN DECA METERS
115   A HE.RH.TOTAL.CPE,      **IN METERS
116   A HE.RH.ROUND.CPE,      **IN METERS
117 BELONGS TO
118   A HE.TB.RH.LIST
119 HAS
120   A P.HE.TB.LIST,
121   A S.HE.TB.LIST,
122   A M.HE.TB.LIST
123
124 EVERY HE.RANGE,HACK,ENVIRONMENT,POSTURE,FUZE HAS
125   AN REPF.LA.PERS          **IN SQM
126
127 EVERY HE.RANGE,HACK,TYPE,EQUIPMENT,ENVIRONMENT,FUZE HAS
128   AN RTEF.LA.EQUIP         **IN SQMX10
129
130
131 EVERY IC.MUNITION HAS      ** 829JAN81 SEC
132   A IC.ID,
133   A IC.WEIGHT,
134   A IC.COST,
135   A IC.RELIABILITY,
136   A IC.SUBM.INDEX,
137   A IC.VOLLEY.RAD,
138   A IC.N.SUBM
139
140 EVERY IC.MUNITION, TYPE,RTY HAS      ** 829JAN81 SEC
141   A IC.TB.SLOPE,
142   A IC.TB.INTERCEPT,
143 OWNS
144   A IC.TB.RH.LIST
145 HAS
146   A F.IC.TB.RH.LIST,
147   A L.IC.TB.RH.LIST,
148   A N.IC.TB.RH.LIST
149
150 EVERY IC.RANGE,HACK HAS      ** 829JAN81 SEC
151   A IC.RH.RANGE,
152   A IC.RH.TOTAL.CPE,
153   A IC.RH.ROUND.CPE,
154 BELONGS TO
155   A IC.TB.RH.LIST
156 HAS

```

Figure III.2.19 (cont)

```

LINE  CACI SINSRIPT II.2  1100 SERIES  RELEASE 6.3

157      A P.IC.TH.RH.LIST,
158      A S.IC.TH.RH.LIST,
159      A P.IC.TH.RH.LIST
160
161  EVERY MISSION HAS
162      A PN.NAME
163
164  EVERY PDB.RNG.HACK HAS
165      A PDB.RN.HANGE
166  BELONGS TO
167      A MPDB.RH.LIST
168  HAS
169      A P.MPDB.RH.LIST,
170      A S.MPDB.RH.LIST,
171      A W.MPDB.RH.LIST
172
173  EVERY POSTURE HAS
174      A PT.NAME
175
176  EVERY SIDE.GROUPING OWNS
177      A UNIT.SET
178  HAS
179      A F.UNIT.SET,
180      A L.UNIT.SET,
181      A N.UNIT.SET
182
183  EVERY SUBDIVISION HAS
184      A SD.NAME          **SUBDIVISION NAME
185
186  EVERY GROUPING HAS
187      A GP.NAME
188  OWNS
189      A GP.CAT.SET
190  HAS
191      A F.GP.CAT.SET,
192      A L.GP.CAT.SET,
193      A N.GP.CAT.SET
194
195  EVERY SIDE.SECTOR HAS
196      A SS.SEAR
197  OWNS
198      A SS.SET
199
200  EVERY SIDE.KILLER.VICTIM HAS
201      A KV.CEN.WPH.NO,
202      A KV.AMPO.CONSUMED,
203      A KV.INITIAL.DENSITY,
204      A KV.EQ.ID
205
206  THE SYSTEM HAS
207      A COUNTER
208  AND OWNS

```

Figure III.2.19 (cont)

L101 CACI SINGSCRIPT 11.0 1100 SERIES RELEASE 6.3

```

209 A BATTLE.SIT,
210 A MPDOL.RH.LIST
211 EVERY TYPE.BTRY HAS
212 A TR.NAME, **ISS, MIN, ETC
213 A TR.HOW.EQ.ID,
214 A TR.H.HOW,
215 A TR.MIN.HOW,
216 A TR.MAX.FERL.FATE, **ROUNDS.MIN*10
217 A TR.SUST.FIRE.RATE, **IN ROUNDS-PER-MINUTE X 10
218 A TR.MAX.WANGE,
219 A TR.MAX.RAP.RANGE, **IN DECA METERS
220 A TR.SFAIL.HEAF.RNDS, **IN ROUNDS
221 A TR.LFAIL.HEAF.RNDS, **IN ROUNDS
222 A TR.SFAIL.REPAIR, **IN HOURS * 10.
223 A TR.LFAIL.REPAIR, **IN HOURS * 10.
224 A TR.SUPPRESS.T.ML, **IN MINUTES
225 A TR.MIN.PREP, **MIN TIME TO PREP A MEN IN MIN*10
226 A TR.MAX.PREP, **MAX TIME TO PREP A MEN IN MIN*10
227 A TR.MIN.FLYA, **IN DECADECMETERS
228 A TR.MAX.FLYA, **IN DECADECMETERS
229 A TR.MARCH.OFFEN, **IN MINUTES
230 A TR.OCCUPY **IN MINUTES
231 OWNS
232 A TR.IN.LIST **TYPE MUNITION POINTERS
233 HAS
234 A F.TR.TY.LIST,
235 A L.TR.TY.LIST
236
237 EVERY TYPE.BTRY,TR...FM HAS
238 A TR.MA.THRESHOLD
239
240 EVERY TYPE.EQUIPMENT, ENVIRONMENT, SUBMUNITION HAS
241 A TES.LA.EQUIP **L A AGAINST EQUIPMENT-50 METERS
242
243 EVERY TYPE.EQUIPMENT OWNS
244 A TE.SLT
245
246 EVERY TYPE.UNIT HAS
247 A TU.LEVEL,
248 A TU.CAT, **PTR TO CATEGORY
249 A TU.MIL.WORTH,
250 A TU.FREQ, **PERCENT OF UNITS IN FORCE THAT ARE
251 A TU.RADIUS, **IN METERS
252 A TU.PRIN.TE,
253 A TU.CRIT.NO, **NUMBER OF CRITICAL PIECES OF EQ
254 A TU.MOV.RATE, **BASIC MV PT FOR UNIT IN MDM PER HR
255 A TU.SIDE,
256 A TU.SUP.PRIORITY,
257 A TU.OPP.PRIORITY
258 OWNS
259 A TU.TE.LIST,
260 A TU.NTC.SET

```

Figure III.2.19 (cont)

LINE CACI SINCRIPT II.6 1107 SCRIPTS RELEASE 6.3

```

261 BELONGS TO
262   A CT.TU.SET
263 HAS
264   A F.TU.TE.LIST,
265   A L.TU.TE.LIST,
266   A F.TU.NTE.SET,
267   A L.TU.NTE.SET,
268   A P.CT.TU.SET
269
270 EVERY TYPE.WEAPON HAS
271   A T.W.NAME,
272   A T.W.RATE.OF.FIRE,
273   A T.W.NO.SENSORS,
274   A T.W.ROUND.VELOCITY,
275   A T.W.ROUND.WEIGHT,
276   A T.W.RND.FIRE,
277   A T.W.MAX.RANGE,
278   A T.W.MIN.RANGE,
279   A T.W.BASIC.LOCAT,
280   A T.W.PR.PTR,
281   A T.W.NITE.FAC,
282   A T.W.FIRE.OTH.PTR
283
284 EVERY UNIT HAS
285   A UN.TU.NOS,
286   A UN.TYPE.UNIT,
287   A UN.X.COORD,
288   A UN.X.GRID,
289   A UN.Y.COORD,
290   A UN.Y.GRID,
291   A UN.RADIUS,
292   A UN.LAST.PTRY.CHG,
293   A UN.TIME.LAST.MOVE,
294   A UN.PARENT,
295   A UN.MISSION,
296   A UN.STATUS,
297   A UN.COLOR,
298   A UN.POSITION.INDEX,
299   A UN.BATTAL.INDEX,
300   A UN.BTRY.INDEX,
301   A UN.PTR
302 AND OWS
303   A UN.EQUIP.LIST,
304   A UN.SUS.LIST,
305   A UN.PATH,
306   A UN.SEGMENT.LIST,
307   A UN.LOS.LIST,
308   A UN.SENSOR.LIST
309 MAY BELONG TO
310   A SS.SET,
311   A UN.SUN.LIST,
312   A FR.UNIT.SET

```

\*\*IN ROUNDS PER MINUTE  
 \*\*IN M/M PER SECOND  
 \*\*LRS 17MAY79 1GLM  
 \*\*17MAY79 1GLM  
 \*\*IN METERS CHG 2PUK 220AUG79  
 \*\*IN METERS CHG 2PUK AUG 79  
 \*\*CHG 2PUK 114AUG79 , 25DEC79  
 \*\*019/4JLR  
 \*\* 25DEC79 1RGR  
 \*\*019/4JLR  
 \*\*2PUK 20AUG79 UNIT NOS SAVE SPACE  
 \*\*TYPE UNIT  
 \*\*IN HEXADLCAMETERS  
 \*\*IN HEXADLCAMETERS  
 \*\*IN HEXADLCAMETERS  
 \*\*IN HEXADLCAMETERS  
 \*\*RADIUS IN METERS  
 \*\*TIME LAST BTRY VOL(\*IN X 100)  
 \*\*TIME SINCE THE UNIT WAS LAST MOVED  
 \*\*PARENT UNIT INDEX  
 \*\*ATTACK, DEFEND, ETC  
 \*\*ADVANCING, WITHDRAWING, OVERWATCH  
 \*\*BLUE OR RED  
 \*\* LINK OF PATH IN CLOSE CMWT  
 \*\* 25DEC79 1RGR  
 \*\* 25DEC79 1RGR

Figure III.2.19 (cont)

LINE CACT SIMSCRIPT II.5 1100 SERIES RELEASE 6.3

```

313 RELO'GS TO
314   A UNIT.SET
315 HAS
316   A F.UN.EQUIP.LIST,
317   A L.UN.EQUIP.LIST,
318   A F.UN.SUB.LIST,
319   A L.UN.SUB.LIST,
320   A F.UN.PATH,
321   A L.UN.PATH,
322   A F.UN.SEGMENT.LIST,
323   A L.UN.SEGMENT.LIST,
324   A F.UN.LOS.LIST,
325   A L.UN.LOS.LIST,
326   A F.UN.SENSOR.LIST,
327   A L.UN.SENSOR.LIST,
328   A F.UN.XM.LIST,
329   A L.UN.XM.LIST,
330   A P.SS.SET,
331   A S.SS.SET,
332   A P.UN.SUB.LIST,
333   A S.UN.SUB.LIST,
334   A P.FR.UNIT.SET,
335   A S.FR.UNIT.SET,
336   A P.AO.DET.TGT.LIST,
337   A S.AO.DET.TGT.LIST,
338   A P.AR.DET.TGT.LIST,
339   A S.AR.DET.TGT.LIST,
340   A P.UNIT.SET,
341   A S.UNIT.SET,
342   A N.UN.EQUIP.LIST,
343   A N.UN.SUB.LIST,
344   A N.UN.PATH,
345   A N.UN.SEGMENT.LIST,
346   A N.UN.LOS.LIST,
347   A N.UN.SENSOR.LIST,
348   A N.SS.SET,
349   A N.UN.SUB.LIST,
350   A M.FR.UNIT.SET,
351   A M.AO.DET.TGT.LIST,
352   A M.AR.DET.TGT.LIST,
353   A M.UNIT.SET
354
355 EVERY UNIT,ENVIRONMENT HAS
356   A UN.ENVIR.FRAC1
357
358 ***** SECTION 3 *****TEMPORARY ENTITIES *****
359
360 TEMPORARY ENTITIES
361
362 EVERY BATTLE HAS
363   A PTL.TIME.OF.DAY,
364

```

\*\*17MAY79 16LM  
\*\*17MAY79 16LM

\*\*20F UNIT IN ENVIRONMENT

Figure III.2.19 (cont)

LINE CACI SINGSCRIPT 11.1 1100 SERIES RELEASE 6.3

```

365     A ETL.TERRAIN.TYPE,
366     A ETL.FIELD,
367     A ETL.BL.UNIT,
368     A ETL.RED.UNIT,
369     A ETL.WIDTH
370 AND BELONGS TO
371     A EATILE.SET
372 HAS
373     A P.PATILE.SET,
374     A S.PATILE.SET,
375     A N.PATILE.SET,
376     A F.PTL.FORCE.SET,
377     A L.PTL.FORCE.SET,
378     A N.PTL.FORCE.SET
379
380 EVERY FORCE HAS
381     A DECISION.POINT,
382     A FR.CRIT.NO,
383     A FR.MISSION,
384     A FR.SIDE
385 MAY OWN
386     A FR.UNIT.SET
387 HAS
388     A F.FR.UNIT.SET,
389     A L.FR.UNIT.SET,
390     A N.FR.UNIT.SET,
391     A P.PTL.FORCE.SET
392
393 EVERY MAN.UNIT HAS
394     A MU.CRIT.NO,
395     A MU.RELIEF.IND,
396     A MU.CUR.ORD,
397     A MU.TF.MEM,
398     A MU.OFFSET.X,
399     A MU.OFFSET.Y,
400     A MU.ID
401 AND OWNS
402     A MU.TF.LIST
403 AND BELONGS TO
404     A MU.TF.LIST
405 HAS
406     A F.MU.TF.LIST,
407     A L.MU.TF.LIST,
408     A N.MU.TF.LIST,
409     A S.MU.TF.LIST,
410     A M.MU.TF.LIST
411
412 EVERY POINT HAS
413     A P.X,
414     A P.Y
415 AND BELONGS TO
416     AN UN.PATH

```

Figure III.2.19 (cont)

LIAL CACI SIMSCRIPT II.5 1100 SERIES RELEASE 6.3

```

417 HAS
418 A P.UN.PATH,
419 A S.UN.PATH,
420 A M.UN.PATH
421
422 EVERY SEGMENT HAS
423     A SEG.LENGTH,
424     A SEG.UNIT,
425     A SEG.TYPE
426 AND BELONGS TO
427     A UN.SEGMENT.LIST
428 HAS
429 A P.UN.SEGMENT.LIST,
430 A S.UN.SEGMENT.LIST,
431 A M.UN.SEGMENT.LIST
432
433 EVERY TH.TM.LINK HAS
434     A TB.TM.RAP,
435     A TB.TM.CLASS,
436     A TB.TM,
437     A TB.TM.FIRED
438 BELONGS TO
439     A TH.TM.LIST
440 HAS
441     A P.TB.TM.LIST,
442     A S.TB.TM.LIST,
443     A M.TB.TM.LIST
444
445 EVERY TU.NTE.LINK HAS
446     A TU.NTE.ID
447 BELONGS TO
448     A TU.NTE.SET
449 HAS
450 A P.TU.NTE.SET,
451 A S.TU.NTE.SET,
452 A M.TU.NTE.SET
453
454 EVERY TU.TE.LINK HAS
455     A TU.TE.ID,
456     A TU.CRITICAL.EQUIP.INDIC,
457     A TU.TE.QUANT
458 BELONGS TO
459     A TU.TE.LIST
460 HAS
461 A P.TU.TE.LIST,
462 A S.TU.TE.LIST,
463 A M.TU.TE.LIST
464
465 EVERY US.LINK HAS
466     A US.SENSOR.TYPE,
467     A US.MODEL,
468     A US.UNIT,

```

```

**ICRAP,UNONRAP
**HE,ICH,PGM,ILLUM,SMOKE
**INDEX TO TYPE MUNITION

```

Figure III.2.19 (cont)

LIFE CACI SIMSCRIPT II.F 1100 SERIES RELEASE 6.3

```
469      A US.FDC,
470      A US.ID,
471      A US.STATUS
472 BELONGS TO
473      A UN.SENSOR.LIST
474 HAS
475      A P.UN.SENSOR.LIST,
476      A S.UN.SENSOR.LIST,
477      A M.UN.SENSOR.LIST
478
479 EVERY VISIBLE.UNIT HAS
480      A VU.POINTER,
481      A VU.STATUS
482 AND BELONGS TO
483      A U.LOS.LIST
484 HAS
485      A P.UN.LOS.SET,
486      A S.UN.LOS.SET,
487      A M.UN.LOS.SET
488
489 EVERY UE.LINK HAS
490      A UE.ID,
491      A UE.CRITICAL.EQUIP.INDIC,
492      A UE.QUANT
493 BELONGS TO
494      A UN.EQUIP.LIST
495 OWNS
496      A UE.WEAPON.SET
497 HAS
498      A P.UN.EQUIP.LIST,
499      A S.UN.EQUIP.LIST,
500      A M.UN.EQUIP.LIST,
501      A F.UE.WEAPON.SET,
502      A L.UE.WEAPON.SET,
503      A N.UE.WEAPON.SET
504
505 EVERY WEAPON HAS
506      A WPN.ID,
507      A WPN.STATUS,
508      A HC.WPN.TYPE,
509      A WPN.QUANTITY,
510      AND SOME WPN.ROUNDS.REMAINING
511 AND BELONGS TO
512      A UE.WEAPON.SET
513 HAS
514      A P.UE.WEAPON.SET,
515      A S.UE.WEAPON.SET,
516      A M.UE.WEAPON.SET
517
518 ***** SECTION 4 ***** PROCESSES *****
519
520
```

Figure III.2.19 (cont)

```

LINE  CACI SIMSCRIPT II.F  1100 SERIES  RELEASE 6.3

521
522 ***** SECTION 5 *****DEFINITIONS *****
523
524 ** A. REAL VARIABLES
525 DEFINE
526     RIL.TIME.OF.DAY,
527     TC.TB.INTERCEPT,
528     TC.TB.SLOPE,
529     UN.TIME.LAST.MOVE,
530     TA.NITE.FAC
531 AS REAL VARIABLES
532
533 ** B. SIGNED INTEGER VARIABLES
534 DEFINE
535     FLUC.UNIT.CNTP,
536     DEMUG,
537     DATA.ERROR,
538     TL.PK.PTR,
539     MU.OFFSET.X,
540     MU.OFFSET.Y,
541     UN.TYPE.UNIT,
542     UN.X.COORD,
543     UN.Y.COORD,
544     UN.X.GRID,
545     UN.Y.GRID,
546     UN.RADIUS,
547     UN.PARENT,
548     HE.RH.RANGE,
549     HE.RH.TOTAL.CPF,
550     HE.RH.ROUND.CPF,
551     TB.HOW.FC-ID,
552     TE.MAX.FIRE.RATE,
553     TB.SUST.FIRE.RATE,
554     TE.MAX.RANGE,
555     TE.SFAIL.MEAN.PIDS,
556     TE.LFAIL.MEAN.RIDS,
557     TB.SFAIL.REPAID,
558     TE.LFAIL.REPAIR,
559     TE.SUPPRESS.TIME,
560     TE.PIN.PROP,
561     TE.MAX.PREP,
562     TE.HIT.FEBA,
563     TB.MAX.FEBA,
564     TU.FREQ,
565     TU.RADIUS,
566     TU.CPIT.NO,
567     TU.MGV.RATE,
568     TW.RATE.OF.FIRE,
569     TW.ROUND.VELOCITY,
570     TA.MAX.RANGE,
571     TB.MIN.RANGE,
572     TB.BASIC.LOAD,

```

Figure III.2.19 (cont)

LINE CACI SIMSCRIPT II.5 1100 SERIES RELEASE 6.3

```

573     DECISION.POINT,
574     FR.CRIT.NO,
575     P.X,
576     P.Y,
577     SEG.LENGTH,
578     SEG.UNIT,
579     SEG.TYPE,
580     TU.CRITICAL.EQUIP.INDIC,
581     TU.TE.QUANT,
582     UL.QUANT
583 AS SIGNED INTEGER VARIABLES
584
585 ** C. ALPHA VARIABLES
586 DEFINE
587     CT.NAME,
588     F1.NAME,
589     F2.NAME,
590     CF.NAME,
591     ME.ID,
592     IC.ID,
593     MH.NAME,
594     PT.NAME,
595     SM.NAME,
596     TB.NAME,
597     TU.LEVEL,
598     TW.NAME,
599     TH.TM.CLASS
600 AS ALPHA VARIABLES
601
602 ** E. SETS
603     DEFINE TU.TE.LIST AS A LIFO SET
604     DEFINE UN.EQUIP.LIST AS A LIFO SET
605     DEFINE SS.SET AS A SET RANKED BY UN.X.COORD
606     DEFINE TB.TM.LIST AS A FIFO SET
607     DEFINE UNIT.SET AND UN.SUP.LIST AS FIFO SETS
608         WITHOUT P,N,AND M ATTRIBUTES
609         WITHOUT FF,FA,FB AND R ROUTINES
610
611 ***** SECTION 6 *****SUBSTITUTIONS *****
612
613 **SUBL SUBSTITUTIONS
614 DEFINE ENDPREABLE TO MEAN END
615 DEFINE ENDRHAIN TO MEAN END
616 DEFINE EXITMAIN TO MEAN STOP
617 DEFINE LNRROUTINE TO MEAN END
618 DEFINE EXITROUTINE TO MEAN RETURN
619 DEFINE ENDFUNCTION TO MEAN END
620 DEFINE EXITFUNCTION TO MEAN RETURN
621 DEFINE LEFTROUTINE TO MEAN LEFT ROUTINE
622 DEFINE UNLEFTROUTINE TO MEAN END
623 DEFINE EXITLEFTROUTINE TO MEAN RETURN
624 DEFINE ENDEVENT TO MEAN END

```

Figure III.2.19 (cont)

```

LINE  CACI SINSRIPT II.5  1100 SERIES  RELEASE 6.3

625  DEFINE EXITEVENT TO MEAN RETURN
626  DEFINE ENDPPROCESS TO MEAN END
627  DEFINE EXITPROCESS TO MEAN RETURN
628  DEFINE ELSEIF TO MEAN ELSE IF
629  DEFINE ENDIF TO MEAN ALWAYS
630  DEFINE ENLOOP TO MEAN REPEAT
631  DEFINE EXITLOOP TO MEAN LEAVE
632  DEFINE LOOP TO MEAN RESUME SUBSTITUTION
633
634  **CONSTANT INTEGER DEFINITIONS
635  DEFINE FALSE TO MEAN 0
636  DEFINE TRUE TO MEAN 1
637  DEFINE BLUE TO MEAN 2
638  DEFINE RED TO MEAN 1
639  DEFINE HIT TO MEAN 1
640  DEFINE MISS TO MEAN 2
641  DEFINE MANLUVER TO MEAN 1
642  DEFINE ARTILLERY TO MEAN 2
643  DEFINE SUPPORT TO MEAN 3
644  DEFINE YES TO MEAN 1
645  DEFINE CPTAVN TO MEAN 4
646
647  ***** SECTION 7 *****COMPUTATION DIRECTIVES *****
648
649  ENDPREAMPL

```

Figure III.2.19 (cont)

THIS PAGE IS INTENTIONALLY LEFT BLANK

LINE CACI SIMSCRIPT II.5 1100 SERIES RELEASE 6.3

```

1  PATH      ** 27 MAPRO #MAIN
2  NORMALLY MODE IS INTEGER
3  LET MAY.DBANK.V = 100000
4  LET N.SIDE=2
5  CREATE EVERY SIDE
6  LET N.GROUPING=3
7  CREATE EVERY GROUPING
8  LET CP.NAME(1)= MAPLOVER
9  LET CP.NAME(2)= ARTILLERY
10 LET CP.NAME(3)= SUPPORT
11 LET DEBUG = TRUE
12 LET DATA.ERROR = FALSE
13 CALL CAT.TU.INPUT
14 PRINT 1 LINE WITH USED.DBANK.V THUS
*** DBANK AFTER CAT.TU.INPUT = *****
15 CALL EQ.TE.INPUT
16 PRINT 1 LINE WITH USED.DBANK.V THUS
*** DBANK AFTER EQ.TE.INPUT = *****
17 CALL UNIT.INPUT
18 PRINT 1 LINE WITH USED.DBANK.V THUS
*** DBANK AFTER UNIT.INPUT = *****
19 CALL P.E.H.INPUT
20 PRINT 1 LINE WITH USED.DBANK.V THUS
*** DBANK AFTER P.E.H.INPUT = *****
21 CALL TP.INPUT
22 PRINT 1 LINE WITH USED.DBANK.V THUS
*** DBANK AFTER TP.INPUT = *****
23 CALL BTRY.INPUT
24 PRINT 1 LINE WITH USED.DBANK.V THUS
*** DBANK AFTER BTRY.INPUT = *****
25 CALL HUNS.INPUT
26 PRINT 1 LINE WITH USED.DBANK.V THUS
*** DBANK AFTER HUNS.INPUT = *****
27 CALL SUMM.INPUT
28 PRINT 1 LINE WITH USED.DBANK.V THUS
*** DBANK AFTER SUMM.INPUT = *****
29 CALL HE.LA.INPUT
30 PRINT 1 LINE WITH USED.DBANK.V THUS
*** DBANK AFTER HE.LA.INPUT = *****
40 LIST N.UNIT,N.TYPE,EQUIPMENT
41 CALL WARE.PRTY
42 ENDMAIN

```

Figure III.2.20

```

LINE  CACI SIMSCRIPT II.5  1100 SERIES  RELEASE 6.3

1  ROUTINE CAT.TU.INPUT  ** NCAT1TUSINPUT
2  NORMALLY MODE IS INTEGER
3  DEFINE CATEGORY, TYPE.UNIT, GROUPING, TU.TE.LINK, TU.TE.LINK
4  AS INTEGER VARIABLES
5  IF DEBUG = TRUE, PRINT 1 LINE THUS
= = = CAT.TU.INPUT = = =
7  ENDIF
8  READ N.CATEGORY
9  CREATE EVERY CATEGORY
10 LOOP FOR EACH CATEGORY DO THIS
11 READ
12 CT.NAME(CATEGORY),
13 CT.MIN.FEBA(CATEGORY),
14 CT.GROUP(CATEGORY)
15 LET CT.MIN.FEBA(CATEGORY) = CT.MIN.FEBA(CATEGORY) * 10. / 16.
16 FILL CATEGORY IN GF.CAT.SET(CT.GROUP(CATEGORY))
17 **PRINT 1 LINE THUS
18 **= = = IN FIRST LOOP = = =
19 ENDLOOP
20 READ N.TYPE.UNIT
21 CREATE EVERY TYPE.UNIT
22 LOOP FOR EACH TYPE.UNIT DO THIS
23 READ
24 TYPE.UNIT,
25 TU.LEVEL(TYPE.UNIT),
26 TU.CAT(TYPE.UNIT),
27 TU.MIL.WORTH(TYPE.UNIT),
28 TU.MOV.RATE(TYPE.UNIT),
29 TU.PRINTE(TYPE.UNIT),
30 TU.RADIUS(TYPE.UNIT),
31 TU.SIDE(TYPE.UNIT),
32 TU.SUP.PRIORITY(TYPE.UNIT),
33 TU.OPP.PRIORITY(TYPE.UNIT)
34 **PRINT 1 LINE THUS
35 **= = = IN SECOND LOOP = = =
36 FILE TYPE.UNIT IN CT.TU.SET(TU.CAT(TYPE.UNIT))
37 LET FLAG = 0
38 LOOP UNTIL FLAG = 1
39 DO THIS
40 CREATE A TU.TE.LINK
41 READ
42 TU.TE.ID(TU.TE.LINK)
43 IF TU.TE.ID(TU.TE.LINK) = 999
44 DESTROY THE TU.TE.LINK
45 LET FLAG = 1
46 ELSE
47 READ TU.TE.QUANT(TU.TE.LINK),
48 TU.CRITICAL.EQUIP.INDIC(TU.TE.LINK)
49 IF TU.CRITICAL.EQUIP.INDIC(TU.TE.LINK)=YES
50 ADD TU.TE.QUANT(TU.TE.LINK) TO TU.CRIT.NO(TYPE.UNIT)
51 ENDIF
52 **PRINT 1 LINE THUS

```

Figure III.2.21

```

LINE  CACI SINSRIPT II.5 1100 SERIES RELEASE 6.3

53 *** == IN THIRD LOOP ==
54 FILE TU.NTE.LINK IN TU.NTE.LIST(TYPE.UNIT)
55 ENDIF
56 ENDOF
57 LET FLAG = 0
58 LOOP UNTIL FLAG = 1
59 DO THIS
60 CREATE A TU.NTE.LINK
61 READ
62 TU.NTE.ID(TU.NTE.LINK)
63 IF TU.NTE.ID(TU.NTE.LINK) = 999
64 DESTROY THE TU.NTE.LINK
65 LET FLAG = 1
66 ELSE
67 FILE TU.NTE.LINK IN TU.NTE.SET(TYPE.UNIT)
68 ENDIF
69 ENDOF
70 ENDOF
71 PRINT 1 LINE WITH TYPE.UNIT THUS ** 423JAN81 SEC
== == LAST TYPE UNIT ***** ==
73 ENDOURTIME

```

Figure III.2.21 (cont)

```

LINE CACI SINSRINT 11.5 1100 SERIES RELEASE 6.3

1 ROUTINE EQ.TE.INPUT ** REQ*TC*INPUT
2 NORMALLY MODE IS INTEGER
3 DEFINE TE.NAME, EQ.NAME AS ALPHA VARIABLES
4 IF DEBUG*TRUE, PRINT 1 LINE THUS
= = = EQ.TE.INPUT = = =
5
6 ENDF
7 READ N.TYPE.EQUIPMENT
8 CREATE EVERY TYPE.EQUIPMENT
9 FOR EACH TYPE.EQUIPMENT READ TE.NAME,
10 TE.PRM.INDIC, TE.PROJECTED.AREA
11 READ N.BLUE.TYPE.EQP, N.RED.TYPE.EQP
12 LET N.EQUIPMENT = N.BLUE.TYPE.EQP + N.RED.TYPE.EQP
13 CREATE EVERY EQUIPMENT
14
15 LOOP FOR EACH EQUIPMENT DO THIS
16 READ
17 EQ.SEQ.NO,
18 EQ.NAME,
19 EQ.TE.PTR(EQUIPMENT),
20 EQ.MAX.SPEED,
21 EQ.PERSONNEL.LOAD
22 ,EQIP.PK.PTR ** 2-22-75 JEN
23 IF EQ.TE.PTR(EQUIPMENT) > 0
24 FILE EQUIPMENT IN TE.SET(EQ.TE.PTR(EQUIPMENT))
25 ELSE
26 CALL ERROR.STOP
27 ENDF
28 READ CEN.EMP.NO
29 ENDOOP
30 PRINT 1 LINE WITH EQ.SEQ.NO THUS ** 12PJAN81 SFC
= = = LAST EQUIP SEQUENCE NUMBER *** = = =
31
32 ENDRoutine

```

Figure III.2.22

LINE CACI SIMSCRIPT II.1 1100 SERIES RELEASE 6.3

```

1  ROUTINE UNIT.INPUT      ** UNITSINPUT
2  NORMALLY MODE IS INTEGER
3  DEFINE SIDE, GROUPING, AND DUMMY AS INTEGER VARIABLES
4  IF DEFUSE=TRUE, PRINT 1 LINE THUS
= = = UNIT.INPUT = = =
5  ENDIF
6  READ N.UNIT
7  RESERVE UNIT.NOS(1) AS N.UNIT
8  CREATE EACH UNIT
9  LOOP FOR EACH UNIT CALLED I DO THIS
10     HEAD UNIT.NOS(I)
11     ENDOLOOP
12     LET I = 0
13     LOOP FOR I = 1 TO N.UNIT DO THIS
14         LET UNIT = I
15         READ UNIT.SEO.NO,
16             UNIT.ID.NO,
17             UN.TYPE.UNIT(I),
18             UN.X.COORD(I),
19             UN.Y.COORD(I),
20             UN.PARENT(I),
21             UN.COLOR(I),
22             UN.RADIUS(I)  **DO NOT USE 333
23             IF UN.COLOR(I) = BLUE
24                 ADD 1 TO BLUE.UNIT.CNTR
25             ENDOF
26         IF UNIT.SEO.NO NE UNIT PRINT 1 LINE WITH UNIT, UNIT.SEO.NO THUS
27         = = = ERROR UNIT **** (IN SEC) HAS SEQUENCE NO **** = = =
28         LET DATA.ERROR = TRUE
29         ENDOF
30         IF UNIT.ID.NO NE UNIT.NOS(I)
31             PRINT 1 LINE WITH I, UNIT.NOS(I), UNIT.ID.NO THUS
32         = = = ERROR UNIT.NOS(****) IS ****, SHOULD BE **** = = =
33         LET DATA.ERROR = TRUE
34         ENDOF
35         IF UN.PARENT(I) > 0,
36             FOR J=1 TO N.UNIT, WITH UNIT.NOS(J)=UN.PARENT(I) FIND THE FIRST CASE
37             IF NONE, SKIP 1 LINE PRINT 1 LINE WITH J, UNIT.NOS(I), UN.PARENT(I) THUS
38             = = = (****) UNIT ****'S PARENT ****, NOT FOUND IN ARRAY = = =
39             LET DATA.ERROR = TRUE
40             ELSE LET SEQ.NO.OF.PARENT=J
41             LET UN.PARENT(I) = SEQ.NO.OF.PARENT
42             ALWAYS ELSE ALWAYS
43             ADD 1 TO TH.FREQ(UN.TYPE.UNIT(I))
44             ADD 1 TO SIDE.TH.TOTAL(UN.COLOR(I))
45             LET UN.X.COORD(I) = 10.*UN.X.COORD(I)/16.
46             LET UN.Y.COORD(I) = 10.*UN.Y.COORD(I)/16.
47             LET UN.STATUS(I) = 3
48             LET SIDE = UN.COLOR(I)
49             LET UN.GROUP = TO.CAT(UN.TYPE.UNIT(I))
50             LET GROUPING = CT.GROUP(UN.GROUP)
51             IF GROUPING=MA.FEUEVER CREATE A MAN.UNIT LET UN.PTR(I)=MAN.UNIT
52

```

Figure III.2.23

LINE CACI SINSRIFT 11.5 1100 SERIES RELEASE 6.3

```

53 READ MU.R.INF.INDIC(UN.PTR(I))
54 ENDF
55 IF MU.UNIT.SET(UNIT)=1 PRINT 1 LINE WITH UNIT.NOS(I), SIDE, GROUPING
  = = = UNIT *** IS FILED IN SIDE *, GROUP *, AN ERROR = = =
56 CALL ERROR.STOP ENDF
57 FILE UNIT IN U.IT.SET(SIDE, GROUPING)
58 IF UN.PARENT(I)>0 FILE UNIT IN UN.EQUIP.LIST(SEQ.NO.OF.PARENT)
59 ENDF
60 LET FLAG = 0
61 LOOP UNTIL FLAG = 1 DO THIS
62   CREATE A UE.LINK CALLED EC
63   READ UE.ID(EC)
64   IF UE.ID(EC) = 999 DESTROY THE UE.LINK CALLED EC
65   LET FLAG = 1
66   ELSE
67     READ UE.QUANT(EC),
68     UE.CRITICAL.EQUIP.INDIC(EC)
69     IF UE.CRITICAL.EQUIP.INDIC(EC)=YES
70     ADD UE.QUANT(EC) TO MU.CRIT.NO(UN.PTR(I))
71     ENDF
72     FILE EC IN THE UN.EQUIP.LIST(I)
73     IF GROUPING=HANDOVER OR GROUPING = CRTAVN
74     LOOP UNTIL WPN.ID(W) = 999 DO THE FOLLOWING
75       CREATE A WEAPON. CALLED W
76       READ WPN.ID(W)
77       IF WPN.ID(W) = 999 DESTROY THE WEAPON CALLED W
78       LEAVE ELSE
79       READ WPN.STATUS(W),
80       WPN.QUANTITY(W)
81       IF GROUPING = CRTAVN
82       READ WPN.WPN.TYPE(W)
83       ENDF
84       FILE W IN THE UE.WEAPON.SET(EC)
85     ENDF
86   ENDF
87 ENDLOOP
88 ENDF
89 ENDF
90 IF DATA.ERROR = TRUE PRINT 1 LINE THUS
  = = = ERRORS IN UNIT DATA = = =
91 CALL ERROR.STOP
92 ENDF
93 RELEASE TIDE.TOTAL
94 PRINT 1 LINE WITH BLUE.UNIT.CNTR THUS
  ==BLUE UNIT COUNT *** ==
95 PRINT 1 LINE WITH UNIT.SEC.NO THUS ** 220JAN61 SEC
  = = = LAST UNIT SEQUENCE NUMBER *** = = =
96 ENDF

```

Figure III.2.23 (cont)

LIFE CACI SIMSCRIPT II.5 1103 SERIES RELEASE 6.3

```

1  ROUTINE P.E.M.INPUT      ** *PT1H*INPUT
2  **THIS ROUTINE INITIALIZES THE POSTURE, ENVIRONMENT, AND MISSION DATA
3  ** THIS VERSION HAS UNIT,ENVIRONMENT FRACTIONS REMOVED &RGP
4  NORMALLY MODE IS INTEGER
5  IF DEBUG=TRUE, PRINT 1 LINE THUS
= = = P.E.M.INPUT = = =
7  ENDIF
8  READ N.POSTURE
9  CREATE EVERY POSTURE
10 FOR EACH POSTURE
11 READ PT.NAME(POSTURE)
12 **LIST ATTRIBUTES OF EACH POSTURE
13 READ N.ENVIRONMENT
14 CREATE EVERY ENVIRONMENT
15 FOR EACH ENVIRONMENT
16 READ EN.NAME(ENVIRONMENT)
17 **LIST ATTRIBUTES OF EACH ENVIRONMENT
18 READ N.MISSION
19 CREATE EVERY MISSION
20 **EACH UNIT HAS A PERCENTAGE IN EACH ENVIRONMENT
21 FOR EACH MISSION
22 READ MN.NAME(MISSION)
23 **LIST ATTRIBUTES OF EACH MISSION
24
25 **THE FOLLOWING LINES OF CODE ARE COMMENTED OUT BECAUSE
26 **A ROUTINE CALLED UNIT.ENVIR IS USED TO COMPUTE THE PERCENT
27 **OF A UNIT IN EACH OF THE ENVIRONMENTS AS REQUIRED BY THE
28 **PROGRAM. REFER TO THAT ROUTINE.
29 **%15JAN80  &RGP
30
31
32 **LOOP FOR EACH UNIT DO THIS
33 **LIST UNIT
34 **LOOP FOR EACH ENVIRONMENT DO THIS
35 **READ UFF          IN PERCENT
36 **LET UN.ENVIR.FRACT(UNIT, ENVIRONMENT) = UFF/2 %10JUN79 %JLR
37 **LIST EN.NAME(ENVIRONMENT), UN.ENVIR.FRACT STORED AS 10,503
38 **UN.ENVIR.FRACT(UNIT, ENVIRONMENT)
39 **ENDLOOP
40 **ENDLOOP
41
42
43 **EACH CATEGORY HAS AN EXPECTED PERCENTAGE IN EACH ENVIRONMENT
44 LOOP FOR EACH ENVIRONMENT DO THIS
45 **LIST EN.NAME(ENVIRONMENT)
46 LOOP FOR EACH CATEGORY DO THIS
47 READ EC.FRACT(ENVIRONMENT, CATEGORY) **IN PERCENT
48 **LIST CT.NAME(CATEGORY),
49 **EC.FRACT(ENVIRONMENT, CATEGORY)
50 ENDLOOP
51 ENDLOOP
52 **EACH CATEGORY IN EACH MISSION HAS A PERCENTAGE IN EACH POSTURE

```

Figure III.2.24

LINE CACI SIMSCRIPT 11.5 1100 SERIES RELEASE 6.3

```
53 **WHICH VARIES DEPENDING ON WHETHER UNIT HAS BEEN WARNED
54 LOOP FOR EACH CATEGORY DO THIS
55 **LIST CT.NAME(CATEGORY)
56 LOOP FOR EACH MISSION DO THIS
57 **LIST MN.NAME(MISSION)
58 LOOP FOR EACH POSTURE DO THIS
59 READ CPM.WARNED.FRACT(CATEGORY, POSTURE, MISSION), **IN PERCENT
60 CPM.UNWARNED.FRACT(CATEGORY, POSTURE, MISSION) **IN PERCENT
61 **LIST PT.NAME(POSTURE),
62 **CPM.WARNED.FRACT(CATEGORY, POSTURE, MISSION),
63 **CPM.UNWARNED.FRACT(CATEGORY, POSTURE, MISSION)
64 ENDOLOOP
65 ENDOLOOP
66 ENDOLOOP
67 ENDRoutine
```

Figure III.24 (cont)

LINE CACI SINGSCRIPT 11.0 1100 SERIES RELEASE 6.3

```

1 ROUTINE TB.INPUT      ** TB.INPUT
2 NORMALLY MODE IS INTEGER
3 PRINT 1 LINE THUS
4 = = TB.INPUT = = =
5 LET N.TB.N.FM = 4
6 CRE L EVERY TB.N.FM
7 READ N.TYPE.BTRY
8 CREATE EVERY TYPE.BTRY
9 LOOP FOR EACH TYPE.BTRY DO THIS
10     READ TB.SEC.NO,
11         TB.NAME(TYPE.BTRY),
12         TB.HOW.EC.ID(TYPE.BTRY),
13         TB.N.HOW(TYPE.BTRY),
14         TB.MIN.HOW(TYPE.BTRY),
15         TB.MAX.FINE.RATE(TYPE.BTRY),
16         TB.SUST.FINE.RATE(TYPE.BTRY),
17         TB.MAX.RANGE(TYPE.BTRY),
18         TB.MAX.RAP.RANGE(TYPE.BTRY),
19         TB.SFAIL.MEAN.RNDS(TYPE.BTRY),
20         TB.LFAIL.MEAN.RNDS(TYPE.BTRY),
21         TB.SFAIL.REPAIR(TYPE.BTRY),
22         TB.LFAIL.REPAIR(TYPE.BTRY),
23         TB.SUPPRESS.TIME(TYPE.BTRY),
24         TB.MIN.PREP(TYPE.BTRY),
25         TB.MAX.PREP(TYPE.BTRY),
26         TB.MIN.FEBA(TYPE.BTRY),
27         TB.MAX.FEBA(TYPE.BTRY),
28         TB.MARCH.ORDER(TYPE.BTRY),
29         TB.OCCUPY(TYPE.BTRY)
30     LET TB.MAX.RANGE(TYPE.BTRY)=TB.MAX.RANGE(TYPE.BTRY) * 10/16.
31     LET TB.MAX.RAP.RANGE(TYPE.BTRY)=TB.MAX.RAP.RANGE(TYPE.BTRY) * 10/16.
32     LET TB.MAX.FEBA(TYPE.BTRY)=TB.MAX.FEBA(TYPE.BTRY) * 10/16.
33     LET TB.MIN.FEBA(TYPE.BTRY)=TB.MIN.FEBA(TYPE.BTRY) * 10/16.
34     LET FLAG = 0
35     LOOP UNTIL FLAG = 1 DO THIS
36         CREATE A TB.TM.LINK
37         READ TB.TM.RAP(TB.TM.LINK),
38             TB.TM.CLASS(TB.TM.LINK),
39             TB.TM(TB.TM.LINK)
40         IF TB.TM.RAP(TB.TM.LINK)=0
41             DESTROY THIS TB.TM.LINK
42             LET FLAG=1
43         ELSE FILE TB.TM.LINK IN TB.TM.LIST(TYPE.BTRY)
44         ENCLIF
45     ENCLLOOP
46     LOOP FOR EACH TB.N.FM DO THIS
47         READ TB.N.THRESHOLD(TYPE.BTRY, TB.N.FM)
48     ENCLLOOP
49 ENCLLOOP
50 PRINT 1 LINE WITH TB.SEC.NO THUS      ** 32PJAN81 SEC
51 = = LAST TYPE BTRY SEQ NO *** = = =
52 ENDRoutine

```

Figure III.2.25

LINE CACI SIMSCRIPT II.5 (110) SERIES RELEASE 6.3

```

1  ROUTINE BTRY.INPUT  **  *BTRY.INPUT
2  NORMALLY MODE IS INTEGER
3  PRINT 1 LINE THUS
4  = BTRY.INPUT = = =
5  READ N.FA.BN
6  SKIP N.FA.BN FIELD
7  READ N.BTRY
8  CREATE EVERY BTRY
9  LET FA.BN=1
10 READ FA.BN.UNIT,
11 DUMMY
12 LOOP FOR EACH BTRY DO THIS
13   'START'
14   IF DUMMY=999
15     READ DUMMY
16     IF DUMMY=999
17       IF BTRY<N.BTRY
18         SKIP 1 LINE
19         PRINT 2 LINES WITH N.BTRY,BTRY THUS
20 NOT ENOUGH BATTERIES HAVE BEEN LISTED IN BATTERY FILE -
21 *** WERE EXPECTED, BUT ONLY *** WERE FOUND.
22     STOP
23   ENDIF
24   GO TO FINISH
25   ENDIF
26   ADD 1 TO FA.BN
27   IF FA.BN>N.FA.BN
28     SKIP 1 LINE
29     PRINT 2 LINES WITH N.FA.BN THUS
30 TOO MANY BATTALIONS HAVE BEEN LISTED IN BATTERY FILE -
31 ONLY *** WERE EXPECTED.
32     STOP
33   ENDIF
34   READ DUMMY
35   GO TO START
36   ENDIF
37   LET BY.BN(BTRY)=FA.BN
38   LET BY.TYPE(BTRY)=DUMMY
39   READ BY.UNIT(BTRY),
40     BY.PGM.CAP(BTRY),
41     DUMMY
42   **LIST BY.UNIT(BTRY) BEFORE .
43   LOOP FOR I = 1 TO N.UNIT, **7/79 PJK UNIT NUMBERING
44     WITH BY.UNIT(BTRY) = UNIT.NOS(I),
45     FIND THE FIRST CASE
46     IF NONE, SKIP 1 LINE
47     PRINT 1 LINE WITH BTRY, BY.UNIT(BTRY) THUS
48 BY.UNIT(****)*** NOT FOUND IN READING BTRY.INPUT.
49     **LET ERROR = 1
50   ENDIF
51   LET BY.UNIT(BTRY) = 1
52 ENDLOOP

```

Figure III.2.26

LIFE CACI SIMSCRIPT II.5 1100 SERIES RELEASE 5.3

```
53 READ DUMMY
54 IF DUMMY NE 999
55   SKIP 1 LINE
56   PRINT 1 LINE THUS
MORE DATA THAN EXPECTED APPEARS IN THE BATTERY FILE.
58   ENDIF
59   'FINISH'
60   IF FA.BNKN.FA.BN
61     SKIP 1 LINE
62     PRINT 2 LINES WITH N.FA.BN.FA.BN THUS
NOT ENOUGH BATTALIONS HAVE BEEN LISTED IN BATTERY FILE -
*** WERE EXPECTED, BUT ONLY *** WERE FOUND.
65     STOP
66   ENDIF
67   ENDROUTINE
```

Figure III.2.26 (cont)

LINE CACI SIMSCRIPT II.5 1100 SERIES RELEASE 6.3

```

1 ROUTINE MUNS.INPUT ** MUNS.INPUT 329JAN81 SEC
2 **THIS ROUTINE READS IN THE ATTRIBUTES OF ALL ICM AND HE MUNITIONS WITH THE
3 **APPROPRIATE FUZES
4 NORMALLY MODE IS INTEGER
5 ** IF DEBUG=TRUE, PRINT 1 LINE THUS
6 ** = = = MUNS.INPUT = = =
7 ** ENDIF
8 READ N.HE.MUNITION, N.IC.MUNITION, N.FUZE
9 CREATE EVERY HE.MUNITION, IC.MUNITION AND FUZE
10 RESERVE HE.TR.N.RH(1,*) AS N.HE.MUNITION BY N.TYPE.BTRY
11 RESERVE IC.TR.N.RH(1,*) AS N.IC.MUNITION BY N.TYPE.BTRY
12 **DETERMINE THE TOTAL NUMBER OF HE RANGE HACKS
13 LOOP FOR EACH TYPE.BTRY DO THIS
14 LOOP FOR EACH HE.MUNITION DO THIS
15 READ HE.TR.N.RH( HE.MUNITION, TYPE.BTRY)
16 LET N.HE.RANGE.HACK = N.HE.RANGE.HACK +
17 HE.TR.N.RH(HE.MUNITION, TYPE.BTRY)
18 ENDOLOOP
19 **DETERMINE THE TOTAL NUMBER OF ICM RANGE HACKS
20 LOOP FOR EACH IC.MUNITION DO THIS
21 READ IC.TR.N.RH( IC.MUNITION, TYPE.BTRY)
22 LET N.IC.RANGE.HACK = N.IC.RANGE.HACK +
23 IC.TR.N.RH(IC.MUNITION, TYPE.BTRY)
24 ENDOLOOP
25 ENDOLOOP
26 CREATE EVERY HE.RANGE.HACK, IC.RANGE.HACK
27 LET COUNT = 0
28 FOR EACH FUZE
29 READ FZ.NAME(FUZE)
30 LOOP FOR EACH HE.MUNITION DO THIS
31 READ
32 HE.IF(HE.MUNITION),
33 HE.WEIGHT(HE.MUNITION),
34 HE.COST(HE.MUNITION),
35 HE.VOLLEY.RAD(HE.MUNITION),
36 HE.ROUND.RAD(HE.MUNITION)
37 LOOP FOR EACH FUZE DO THIS
38 READ FZ.HE.RELY(FUZE, HE.MUNITION) **TIMES 100
39 **LIST FZ.HE.RELY(FUZE, HE.MUNITION)
40 ENDOLOOP
41 **READ ATTRIBUTES OF EACH RANGE HACK (WHICH IS ASSOCIATED WITH
42 ** A SPECIFIC HE MUNITION FIRED BY A SPECIFIC TYPE BATTERY AT A SPECIFIC
43 **RANGE
44 LOOP FOR EACH TYPE.BTRY WHEN HE.TR.N.RH(HE.MUNITION, TYPE.BTRY) > 0 DO THIS
45 LET A = COUNT + 1
46 LET R = COUNT + HE.TR.N.RH(HE.MUNITION, TYPE.BTRY)
47 LOOP FOR RH = A TO R DO THIS
48 FILE THE RH IN HE.TR.RH.LIST(HE.MUNITION, TYPE.BTRY)
49 READ
50 R.RANGE, ** %5DEC79 %RGR
51 HE.RH.TOTAL.CPE(RH),
52 HE.RH.ROUND.CPE(RH)

```

Figure III.2.27

LINE CAGE SIMSCRIPT II.\* 1100 SERIES RELEASE 5.3

```

13 LET HE.RH.RANGE(RH) = R.RANGE*10/16. ** 25DEC79 SNGR
14 ENDOLOOP
15 LET COUNT = COUNT + HE.TB.N.RH(HE.MUNITION, TYPE.BTRY)
16 ENDOLOOP
17 ENDOLOOP
18 LET COUNT = 0
19 LOOP FOR EACH IC.MUNITION DO THIS
20 READ
21 IC.ID(IC.MUNITION),
22 IC.WEIGHT(IC.MUNITION),
23 IC.COST(IC.MUNITION),
24 IC.RELIABILITY(IC.MUNITION), **TIMES 100.
25 IC.SUBM.INDEX(IC.MUNITION),
26 IC.VOLLEY.RAD(IC.MUNITION),
27 IC.N.SUBM(IC.MUNITION)
28 LOOP FOR EACH TYPE.BTRY WITH IC.TB.N.RH(IC.MUNITION, TYPE.BTRY) > 0 DO THIS
29 READ IC.TB.SLOPE(IC.MUNITION, TYPE.BTRY),
30 IC.TB.INTERCEPT(IC.MUNITION, TYPE.BTRY)
31 **LIST IC.TB.SLOPE(IC.MUNITION, TYPE.BTRY),
32 ** IC.TB.INTERCEPT(IC.MUNITION, TYPE.BTRY)
33 LET A = COUNT + 1
34 LET B = COUNT + IC.TB.N.RH(IC.MUNITION, TYPE.BTRY)
35 **READ ATTRIBUTES OF EACH RANGE HACK (WHICH IS ASSOCIATED WITH
36 ** A SPECIFIC IC MUNITION FIRED BY A SPECIFIC TYPE BATTERY AT A SPECIFIC
37 ** RANGE
38 LOOP FOR RH = A TO B DO THIS
39 FILE THE RH IN IC.TB.RH.LIST(IC.MUNITION, TYPE.BTRY)
40 READ
41 R.RANGE, ** 25DEC79 SNGR
42 IC.RH.TOTAL.CPL(RH),
43 IC.RH.POUND.CPL(RH)
44 LET IC.RH.RANGE(RH) = R.RANGE*10/16. ** 25DEC79 SNGR
45 ENDOLOOP
46 LET COUNT = COUNT + IC.TB.N.RH(IC.MUNITION, TYPE.BTRY)
47 ENDOLOOP
48 ENDOLOOP
49 **LIST ATTRIBUTES OF EACH FUZL
50 **LOOP FOR EACH HE.MUNITION DO THIS
51 **LIST ATTRIBUTES OF HE.MUNITION
52 **LOOP FOR EACH TYPE.BTRY DO THIS
53 **LIST ATTRIBUTES OF EACH HE.RANGE.HACK IN
54 **HE.TB.RH.LIST(HE.MUNITION, TYPE.BTRY)
55 **ENDLOOP
56 **ENDLOOP
57 **LOOP FOR EACH IC.MUNITION DO THIS
58 **LIST ATTRIBUTES OF IC.MUNITION
59 **LOOP FOR EACH TYPE.BTRY DO THIS
60 **LIST ATTRIBUTES OF EACH IC.RANGE.HACK IN
61 **IC.TB.RH.LIST(IC.MUNITION, TYPE.BTRY)
62 **ENDLOOP
63 **ENDLOOP
64 RELEASE HE.TB.N.RH, IC.TB.N.RH
65 ENDOURTIME

```

Figure III.2.27 (cont)

LINE CACI SIMSCRIPT (7.5 1100 SERIES RELEASE 6.3

```

1 ROUTINE SUBM.INPUT ** #SUBM.INPUT
2 **THIS ROUTINE READS IN THE DATA SET FOR EACH SUBMUNITION
3 NORMALLY NODE IS INTEGER
4 IF DEBUGTRUE, PRINT 1 LINE THUS
= = = SUBM.INPUT = = =
5   ENDOF
6   DEFINE PERSONNEL TO MEAN 1
7   READ N.SUBMUNITION
8   CREATE EVERY SUBMUNITION
9   FOR EACH SUBMUNITION
10    READ SM.NAME(SUBMUNITION)
11    LOOP FOR EACH SUBMUNITION
12    FOR EACH ENVIRONMENT DO THIS
13    READ ES.RELY(ENVIRONMENT,SUBMUNITION) **TIMES 100
14    **LIST ES.RELY(ENVIRONMENT,SUBMUNITION)
15    ENDOLOOP
16    LOOP FOR EACH SUBMUNITION
17    FOR EACH ENVIRONMENT
18    FOR EACH POSTURE DO THIS
19    READ EPS.LA.PERS( ENVIRONMENT,POSTURE,SUBMUNITION ) **IN SQ.METERS*10
20    **LIST EPS.LA.PERS( ENVIRONMENT,POSTURE,SUBMUNITION )
21    ENDOLOOP
22    LOOP FOR EACH SUBMUNITION
23    FOR EACH ENVIRONMENT
24    FOR EACH TYPE.EQUIPMENT UNLESS TYPE.EQUIPMENT=PERSONNEL DO THIS
25    READ TES.LA.EQUIP( TYPE.EQUIPMENT,ENVIRONMENT,SUBMUNITION)**IN SQ.M.*10
26    **LIST TES.LA.EQUIP( TYPE.EQUIPMENT,ENVIRONMENT,SUBMUNITION)
27    ENDOLOOP
28    PRINT 1 LINE WITH TES.LA.EQUIP(23,3,4) THUS ** 428JAN81 SEC
= = = LAST SUBMUNITION LA AND *** = = =
31 ENDOUTIN

```

Figure III.2.28

LINE CACI SIMSCRIPT 11.5 1100 SERIES RELEASE 6.3

```

1 ROUTINE HE.LA.INPUT      ** 5-31-79 JN - TO LIST DATA      MHESLAINPUT
2 **THIS ROUTINE READS IN LETHALITY DATA FOR EACH HE MUNITION
3 NORMALLY MODE IS INTEGER
4 DEFINE PERSONNEL TO MEAN 1
5 IF DEBUG=TRUE, PRINT 1 LINE THUS
= = = HE.LA.INPUT = = =
7     ENDTF
8     LOOP FOR EACH TYPE.BTRY
9     FOR EACH HE.MUNITION
10    FOR EACH FUZE
11    FOR EACH POSTURE
12    FOR EACH ENVIRONMENT
13    FOR EACH RH IN HE.TG.RH.LIST
14    (HE.MUNITION,TYPE.BTRY)
15    DO THIS
16    ** LIST N.HE.TG.RH.LIST(HE.MUNITION,TYPE.BTRY) 5-31-79 JN INSERT
17    ** LIST TYPE.BTRY,HE.MUNITION,FUZE,POSTURE,ENVIRONMENT,RH 25-31JN
18    READ REPE.LA.PERS **IN SQ.METERS
19    (RH,ENVIRONMENT,POSTURE,FUZE)
20    ** LIST REPE.LA.PERS 5-31-79 JN ACTIVATED
21    ** (RH,ENVIRONMENT,POSTURE,FUZE)
22    ENDOOP
23    LOOP FOR EACH TYPE.BTRY
24    FOR EACH HE.MUNITION
25    FOR EACH FUZE
26    FOR EACH ENVIRONMENT
27    FOR EACH TYPE.EQUIPMENT UNLESS TYPE.EQUIPMENT=PERSONNEL
28    FOR EACH RH IN HE.TG.RH.LIST
29    (HE.MUNITION,TYPE.BTRY)
30    DO THIS
31    ** LIST N.HE.TG.RH.LIST 5-31-79 JN INSERT
32    ** LIST TYPE.BTRY,HE.MUNITION,FUZE,ENVIRONMENT,TYPE.EQUIPMENT,RH 25-31JN
33    READ RTCE.LA.EQUIP **IN SQ.METERS*10
34    (RH,TYPE.EQUIPMENT,ENVIRONMENT,FUZE)
35    **LIST RTCE.LA.EQUIP 5-31-79 JN ACTIVATED
36    ** (RH,TYPE.EQUIPMENT,ENVIRONMENT,FUZE)
37    ENDOOP
38    ENDRoutine

```

Figure III.2.29

LINE CACI SIMSCRIPT II.4 1107 SERIES RELEASE 6.3

```

1 ROUTINE WARF,ARTY ** WARF,ARTY 830JAN81 SEC
2
3 **THIS ROUTINE USES THE COSAGE RESULTS OF RED ARTILLERY FIRING ON
4 **BLUE UNITS, CALCULATES LOSSES TO ITEMS VULNERABLE TO ARTILLERY
5 **FIRE ATTRITION AND PASSES THE RESULTS SO THEY MAY BE POSTED ON
6 **THE UNIT STATUS FILE, THE RESULTS OF THE UNIT STATUS FILE UPDATE,
7 **WHICH EFFECT THE NUMBER OF AVAILABLE TARGETS THEN ARE RETURNED
8 **TO THIS ROUTINE FOR THE NEXT ITERATION.
9
10 NORMALLY MORE IS INTEGER
11 DEFINE TOE AS A 2-DIMENSIONAL REAL ARRAY **25JAN81 SEC
12 DEFINE LA,X,ROUND,RLY,COVER,PI,PAD,?,FRAC,CAS AS REAL VARIABLES
13 DEFINE FIRST,FRAC,CAS,REMAIN,FRAC,CAS AS REAL VARIABLES **930JAN81 SEC
14 DEFINE PERSONNEL TO MEAN 1
15 DEFINE FM,TH,CLASS AS AN ALPHA VARIABLE ** 22RJAN81 SEC
16 DEFINE ROUND,NAME AS AN ALPHA VARIABLE ** 22RJAN81 SEC
17
18 **READ DETAILS OF THE FIRE MISSION FROM COSAGE WARF HISTORY TAPE
19 **FOR THE TARGET UNIT READ NO OF TARGETS AVAILABLE IN EACH CATEGORY
20 **FROM THE UNIT STATUS FILE
21 **COMPUTE THE FRACTIONAL CASUALTIES USING THE APPROPRIATE ALGORITHM
22 **SET
23
24 IF DEBUG = TRUE, PRINT 1 LINE THUS
= = = ENTERING WARF,ARTY = = =
25
26 ENDIF
27 RESERVE TOE AS BLUE,UNIT,CNTR BY 23 ** 25JAN81 SEC
28 FOR I = 1 TO BLUE,UNIT,CNTR DO
29     FOR J = 1 TO 23 DO
30         READ TOE(I,J) ** 22JAN81 SEC
31     ENDOFOR
32 ENDOFOR
33 PRINT 1 LINE WITH TOE(18?,23) THUS ** 22JAN81 SEC
= = = LAST TOE QTY READ **** = = =
34 LET FLAG = 0
35 USE UNIT 4 FOR INPUT
36 LOOP UNTIL FLAG = 1 DO THIS
37     IF DATA IS ENDED
38         LET FLAG=1
39     LEAVE
40 ENDIF
41 READ TIME,V,SIDE,FIRE,PLAN
42 ** PRINT 1 LINE WITH TIME,V,SIDE,FIRE,PLAN THUS
43 ** = = = TIME,V ****,SIDE ***** FIRE,PLAN ***** = = =
44 READ RANGE,FM,FIRED,VOLS,BTRY
45 ** PRINT 1 LINE WITH RANGE,FM,FIRED,VOLS,BTRY THUS
46 ** = = = RANGE ***** FM,FIRED,VOLS ***** BTRY ***** = = =
47 READ BY,UNIT(BTRY),TB,TUBES
48 ** PRINT 1 LINE WITH BY,UNIT,TP,TUBE THUS
49 ** = = = BY,UNIT ***** TB ***** TUBE ***** = = =
50 READ UNIT,TU,CAT
51 ** PRINT 1 LINE WITH UNIT,TU,CAT THUS
52

```

Figure III.2.30

LINE CACI SIMSCRIPT II.5 1100 SERIES RELEASE 6.3

```

53 ** = = = UNIT ***** TO ***** CAT ***** = = =
54 READ FM.TM.CLASS, MUNITION, ROUND, NAME
55 ** PRINT 1 LINE WITH FM.TM.CLASS, MUNITION, ROUND, NAME THUS
56 ** = = = FM.TM.CLASS ***** MUNITION ***** ROUND, NAME ***** = = =
57 READ FUZE, RNG, HACK, ROUND, RELY
58 ** PRINT 1 LINE WITH FUZE, RNG, HACK, ROUND, RELY THUS
59 ** = = = FUZE ***** RNG, HACK ***** ROUND, RELY **, ** = = =
60 READ NUM.ROUNDS, ST, QUANT
61 ** PRINT 1 LINE WITH NUM.ROUNDS, ST, QUANT THUS
62 ** = = = NUM.ROUNDS ***** ST ***** QUANT ***** = = =
63 READ COVER, PI, RAD, 2, MSN
64 ** PRINT 1 LINE WITH COVER, PI, RAD, 2, MSN THUS
65 ** = = = COVER **, **, PI, RAD, 2 ***** **, MSN ***** = = =
66 READ TOTAL, CAS
67 ** PRINT 1 LINE WITH TOTAL, CAS THUS
68 ** = = = TOTAL, CAS ***** = = =
69 ** PRINT 1 LINE WITH TIME, V THUS
DATA READ FOR TIME, V = ***, *****
71 LOOP FOR EACH TYPE, EQUIPMENT CALLED TE, DO THIS
72 LET FRAC, CAS = 0.0
73 LOOP FOR EACH ENVIRONMENT CALLED ENVIR, DO THIS
74 LET FRAC, ENVIR = REAL, F( LC, FRAC( ENVIR, CAT ) ) / 50.
75 IF FM.TM.CLASS = ICM
76 IF TE NE PERSONNEL
77 LET LA = REAL, F( TES, LA, EQUIP( TE, ENVIR, ST ) )
78 / 10.
79 LET X = QUANT * ROUND, RELY * ES, RELY( ENVIR, ST )
80 / 100. * COVER * LA / PI, RAD, 2
81 LET FRAC, CAS = FRAC, CAS + ( 1.0 - EXP, F ( - X ) )
82 * FRAC, ENVIR
83 ** 7779 JLN UN, ENVIR, FRAC, STORP
84 ELSE ** PERSONNEL AS EQ, 503
85 LOOP FOR EACH POSTURE CALLED POST, DO THIS
86 LET LA = REAL, F( EPS, LA, PERSE( ENVIR, POST,
87 ST ) ) / 10
88 LET X = QUANT * ROUND, RELY * ES, RELY(
89 ENVIR, ST ) / 100. * LA
90 * COVER / PI, RAD, 2
91 LET FRAC, CAS = FRAC, CAS + ( 1.0 - EXP, F(
92 -X ) ) * FRAC, ENVIR
93 * REAL, F( CPH, WARNED, FRAC( CAT, POST,
94 MISSION, J ) ) / 100.
95 ENDOLOOP
96 ENDIF
97 ELSE ** MC
98 IF EQ PERSONNEL
99 IF TIME, V * MINUTES, V - UN, LAST, ARTY, ENG( UNIT )
100 > 10. ** 2-6-79 *
101 ** THEN FIRST VULLY HAS TROOPS UNWARNED
102 LET FIRST, FRAC, CAS = 0.0
103 LET REMAIN, FRAC, CAS = 0.0
104 LOOP FOR EACH POSTURE CALLED POST, DO THIS

```

Figure III.2.30 (cont)

LINE CACI SCRIPT 11.1 1100 SERIES RELEASE 5.3

```

105 LET LA = REAL.FI REPT.LA,POST
106 RND.HACK, ENVIR, POST, FUZE 1 )
107 LET FRAC.POST = REAL.FI
108 CPM.UNWARNED.FRAC(CAT, POST, MISSION
109 1 ) / 100.
110 LET FIRST.FRAC.CAS = FIRST.FRAC.CAS +
111 ( ( LA * ROUND.RELY * COVER * TUBES )
112 / PI.RAD.2 ) * FRAC.POST
113 LET FRAC.POST = REAL.FI CPM.WARNED.FRAC
114 ( CAT, POST, MISSION ) / 100.
115 LET X = ( QUANT - TUBES ) * LF
116 * ROUND.RELY * COVER / PI.RAD.2
117 LET REMAIN.FRAC.CAS = REMAIN.FRAC.CAS
118 * ( 1.0 - CPM.FI - X ) * FRAC.POST
119 ENDOOP
120 LET FRAC.CAS = FRAC.CAS + ( FIRST.FRAC.CAS
121 * ( 1.0 - FIRST.FRAC.CAS ) *
122 REMAIN.FRAC.CAS ) * FRAC.ENVIR
123 ELSE **TROOPS ARE WARNED ON FIRST VOLLEY
124 LOOP FOR EACH POSTURE CALL FRAC.POST, DO THIS
125 LET LA = REAL.FI REPT.LA,POST
126 RND.HACK, ENVIR, POST, FUZE 1 )
127 LET FRAC.POST = REAL.FI
128 CPM.WARNED.FRAC(CAT,
129 POST, MISSION ) / 10
130 LET X = QUANT * LA * ROUND.RELY *
131 COVER / PI.RAD.2
132 LET FRAC.CAS = FRAC.CAS + ( 1.0 - EXP.F
133 ( -X ) ) * FRAC.POST * FRAC.ENVIR
134 ENDOOP
135 ENDIF
136 ELSE ** TE NOT PERSONNEL
137 LET LA = REAL.FI REPT.LA,EQUIP RND.HACK, TE,
138 ENVIR, FUZE 1 ) / 10.
139 LET X = QUANT * LA * ROUND.RELY * COVER /
140 PI.RAD.2
141 LET FRAC.CAS = FRAC.CAS + ( 1.0 - EXP.F ( -X ) )
142 * FRAC.ENVIR
143 ENDIF
144 ENDIF
145 ENDOOP
146 SUBTRACT FRAC.CAS * TOE(UNIT,TE) FROM TOE(UNIT,TE)
147 ** ADJUST SEC
148 ** PRINT 1 LINE WITH FRAC.CAS THUS
149 ** = = = FRAC.CAS EQUALS ****.*** = = =
150 ENDOOP
151 ENDOOP
152 LOOP FOR I = 1 TO BLUE.UNIT.CNTR, DO
153 LET UNIT = I
154 FOR J = 1 TO 12, DO
155 LET TC = J
156 WRITE TO(UNIT,TE) AS D(7,2) USING UNIT 3
157 ENDOOP
158 WRITE AS / USING UNIT 3
159 FOR K = 13 TO 23, DO
160 LET TC = K
161 WRITE TO(UNIT,TE) AS D(7,2) USING UNIT 3
162 ENDOOP
163 WRITE AS / USING UNIT 3
164 ENDOOP
165 ENDROUTINE

```

Figure III.2.30 (cont)

```
LINE CACI SIMSCRIPT 11.0 1100 SERIES RELEASE 6.3  
1 ROUTINE ERROR STOP  
2 PRINT 1 LINE THIS  
* * * AN ERROR HAS OCCURRED * * *  
4 STOP  
5 END
```

Figure III.2.31

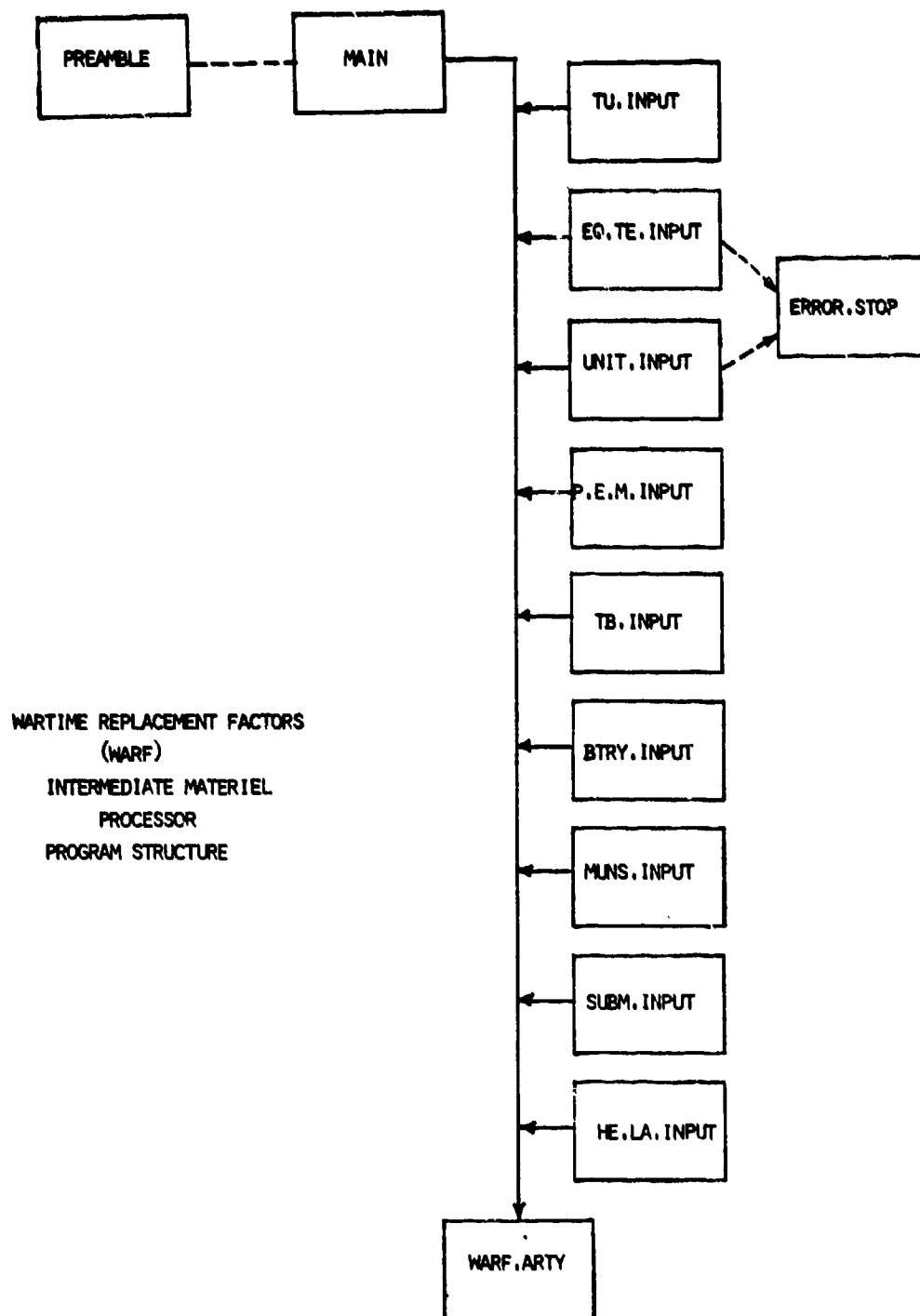


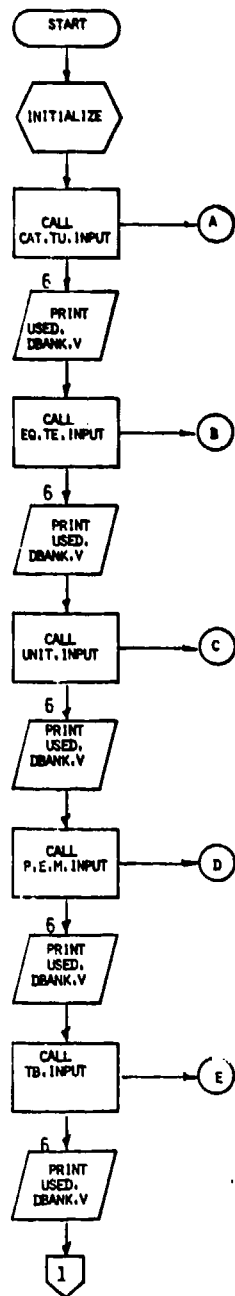
Figure III.2.32

UNCLASSIFIED\*\*\*IMPDI\*\*\*UNCLASSIFIED

1:	0.	6.	7.	8.	9.	111.	500.	600.	600.	40.	50.	60.
2:	7.	3.	34.	35.	678.	298.	389.	4.	57.	34.	9.	
3:	43.	2.	56.	2.	4.	11.	34.	54.	56.	765.	67.	7.
4:	45.	5.	66.	7.	11.	32.	13.	2.	45.	6.	3.	
5:	5.	6.	7.	8.	9.	111.	500.	600.	600.	40.	50.	60.
6:	7.	3.	34.	35.	678.	298.	389.	4.	57.	34.	9.	
7:	-1.106+	1.46	38.27	1.37	3.77	10.34	32.07	50.94	41.01	522.75	45.78	6.60
8:	42.45	4.72	62.26	5.13	7.52	21.67	12.26	1.89	42.45	5.66	2.20	
9:	5.	6.	7.	8.	9.	111.	500.	600.	600.	40.	50.	60.
10:	7.	3.	34.	35.	678.	298.	389.	4.	57.	34.	9.	
11:	43.	2.	56.	2.	4.	11.	34.	54.	56.	765.	67.	7.
12:	45.	5.	66.	7.	11.	32.	13.	2.	45.	6.	3.	
13:	5.	6.	7.	8.	9.	111.	500.	600.	600.	40.	50.	60.
14:	7.	3.	34.	35.	678.	298.	389.	4.	57.	34.	9.	
15:	43.	2.	56.	2.	4.	11.	34.	54.	56.	765.	67.	7.
16:	45.	5.	66.	7.	11.	32.	13.	2.	45.	6.	3.	
17:	5.	6.	7.	8.	9.	111.	500.	600.	600.	40.	50.	60.
18:	7.	3.	34.	35.	678.	298.	389.	4.	57.	34.	9.	
19:	43.	2.	56.	2.	4.	11.	34.	54.	56.	765.	67.	7.
20:	45.	5.	66.	7.	11.	32.	13.	2.	45.	6.	3.	
21:	5.	6.	7.	8.	9.	111.	500.	600.	600.	40.	50.	60.
22:	7.	3.	34.	35.	678.	298.	389.	4.	57.	34.	9.	
23:	43.	2.	56.	2.	4.	11.	34.	54.	56.	765.	67.	7.
24:	45.	5.	66.	7.	11.	32.	13.	2.	45.	6.	3.	
25:	6.9F+15	4.36	5.35	7.77	8.48	99.90	316.05	416.13	397.52	28.37	48.17	55.65
26:	6.12	1.68	21.49	24.27	449.20	211.36	374.75	3.71	49.86	20.98	5.69	
27:	43.	2.	56.	2.	4.	11.	34.	54.	56.	765.	67.	7.
28:	45.	5.	66.	7.	11.	32.	13.	2.	45.	6.	3.	
29:	5.	6.	7.	8.	9.	111.	500.	600.	600.	40.	50.	60.
30:	7.	3.	34.	35.	678.	298.	389.	4.	57.	34.	9.	
31:	43.	2.	56.	2.	4.	11.	34.	54.	56.	765.	67.	7.
32:	45.	5.	66.	7.	11.	32.	13.	2.	45.	6.	3.	
33:	5.	6.	7.	8.	9.	111.	500.	600.	600.	40.	50.	60.
34:	7.	3.	34.	35.	678.	298.	389.	4.	57.	34.	9.	
35:	43.	2.	56.	2.	4.	11.	34.	54.	56.	765.	67.	7.
36:	45.	5.	66.	7.	11.	32.	13.	2.	45.	6.	3.	
37:	5.	6.	7.	8.	9.	111.	500.	600.	600.	40.	50.	60.
38:	7.	3.	34.	35.	678.	298.	389.	4.	57.	34.	9.	
39:	43.	2.	56.	2.	4.	11.	34.	54.	56.	765.	67.	7.
40:	45.	5.	66.	7.	11.	32.	13.	2.	45.	6.	3.	
41:	5.	6.	7.	8.	9.	111.	500.	600.	600.	40.	50.	60.
42:	7.	3.	34.	35.	678.	298.	389.	4.	57.	34.	9.	
43:	43.	2.	56.	2.	4.	11.	34.	54.	56.	765.	67.	7.
44:	45.	5.	66.	7.	11.	32.	13.	2.	45.	6.	3.	
45:	5.	6.	7.	8.	9.	111.	500.	600.	600.	40.	50.	60.
46:	7.	3.	34.	35.	678.	298.	389.	4.	57.	34.	9.	
47:	43.	2.	56.	2.	4.	11.	34.	54.	56.	765.	67.	7.
48:	45.	5.	66.	7.	11.	32.	13.	2.	45.	6.	3.	
49:	5.	6.	7.	8.	9.	111.	500.	600.	600.	40.	50.	60.
50:	7.	3.	34.	35.	678.	298.	389.	4.	57.	34.	9.	
51:	43.	2.	56.	2.	4.	11.	34.	54.	56.	765.	67.	7.
52:	45.	5.	66.	7.	11.	32.	13.	2.	45.	6.	3.	
53:	5.	6.	7.	8.	9.	111.	500.	600.	600.	40.	50.	60.
54:	7.	3.	34.	35.	678.	298.	389.	4.	57.	34.	9.	
55:	43.	2.	56.	2.	4.	11.	34.	54.	56.	765.	67.	7.
56:	45.	5.	66.	7.	11.	32.	13.	2.	45.	6.	3.	
57:	5.	6.	7.	8.	9.	111.	500.	600.	600.	40.	50.	60.

Figure III.2.33

SIDE (2)  
GROUPING (3)  
DEBUG  
DATA: ERROR  
MAX. DBANK. V



MAIN

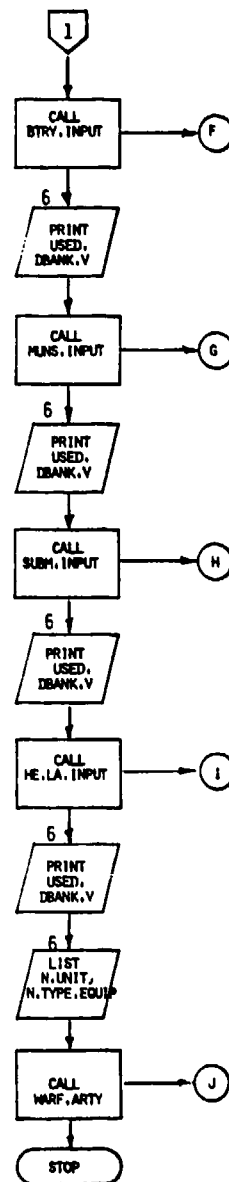


Figure III.2.34

ROUTINE  
CAT.TU.INPUT

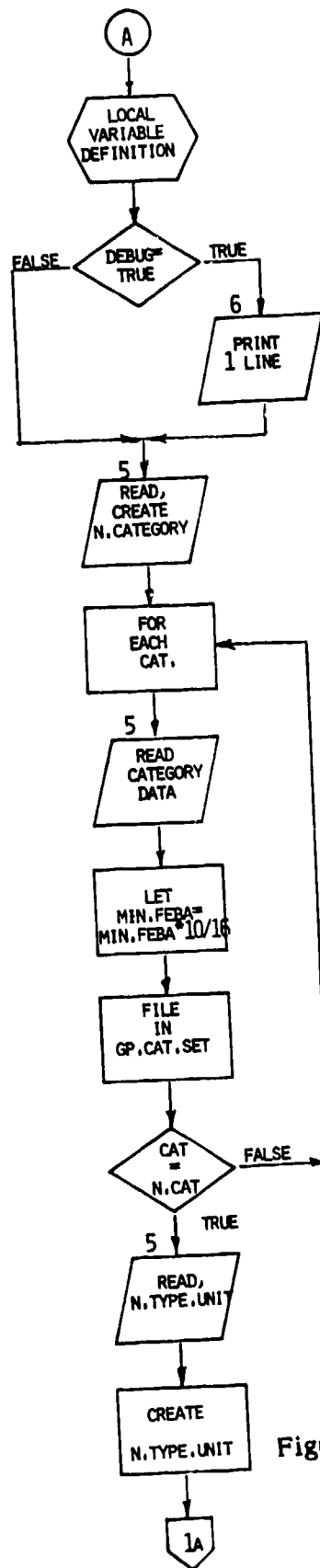


Figure III.2.35

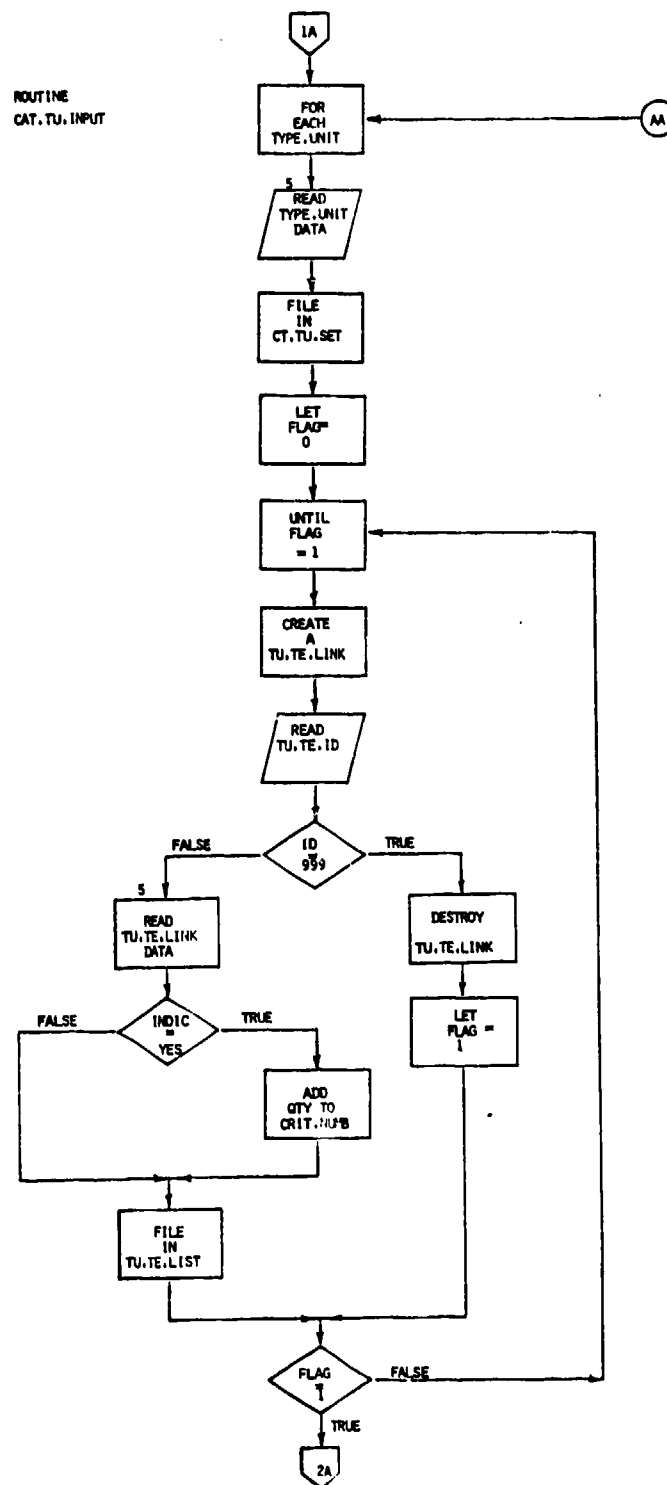


Figure III.2.35 (Cont)

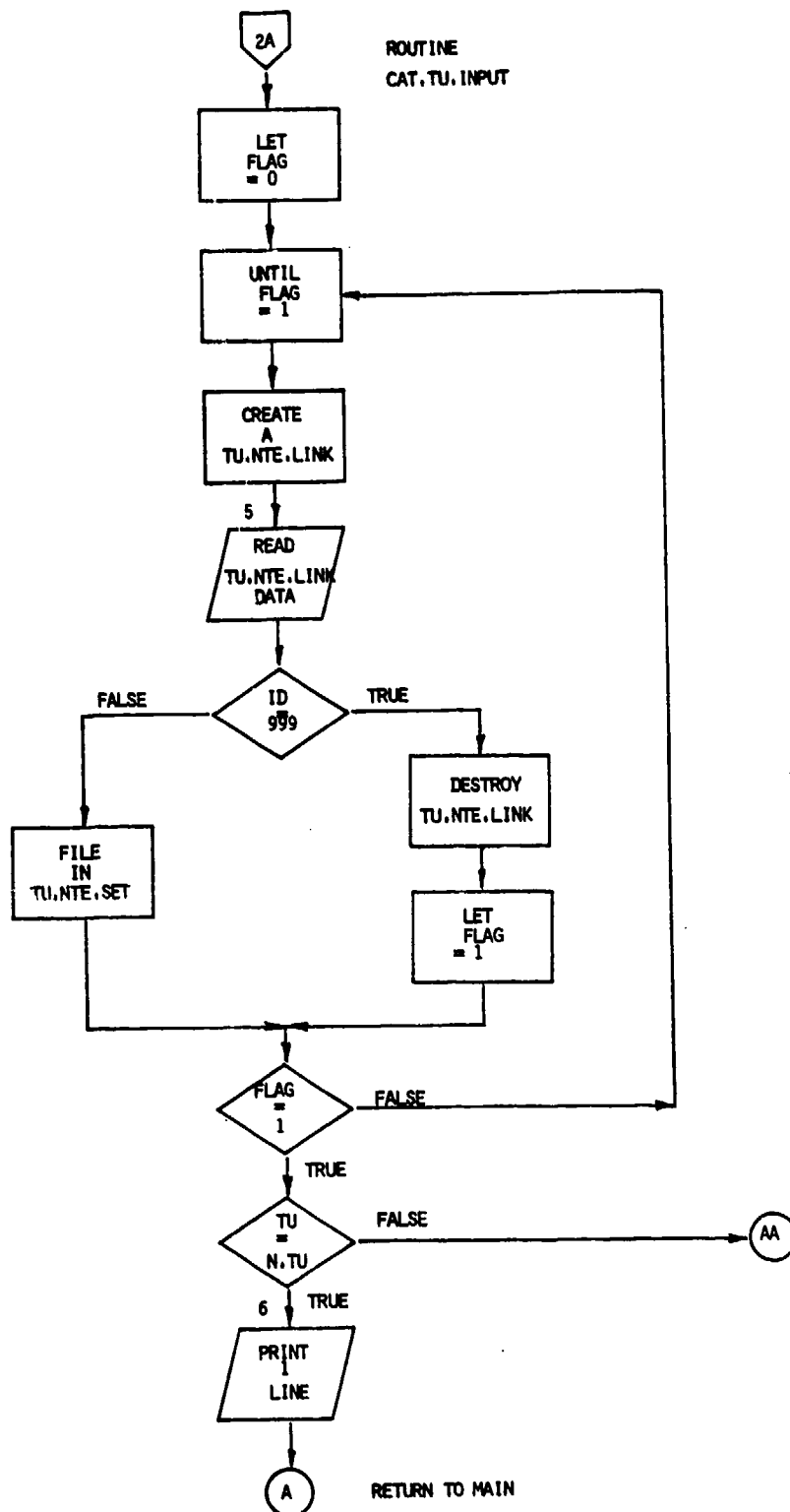


Figure III.2.35 (Cont)

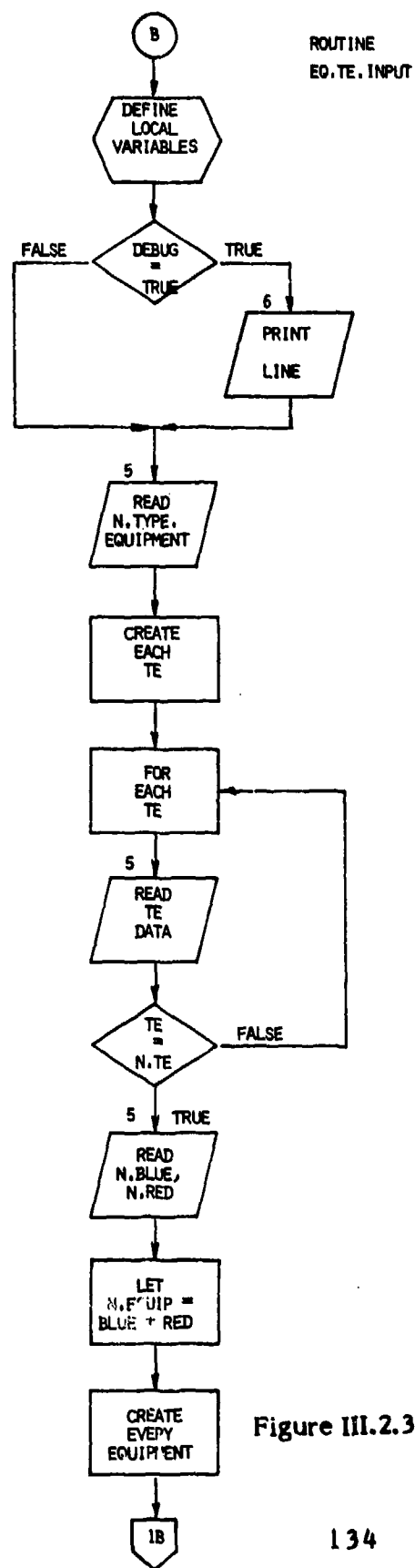


Figure III.2.36

ROUTINE  
EQ.TE.INPUT

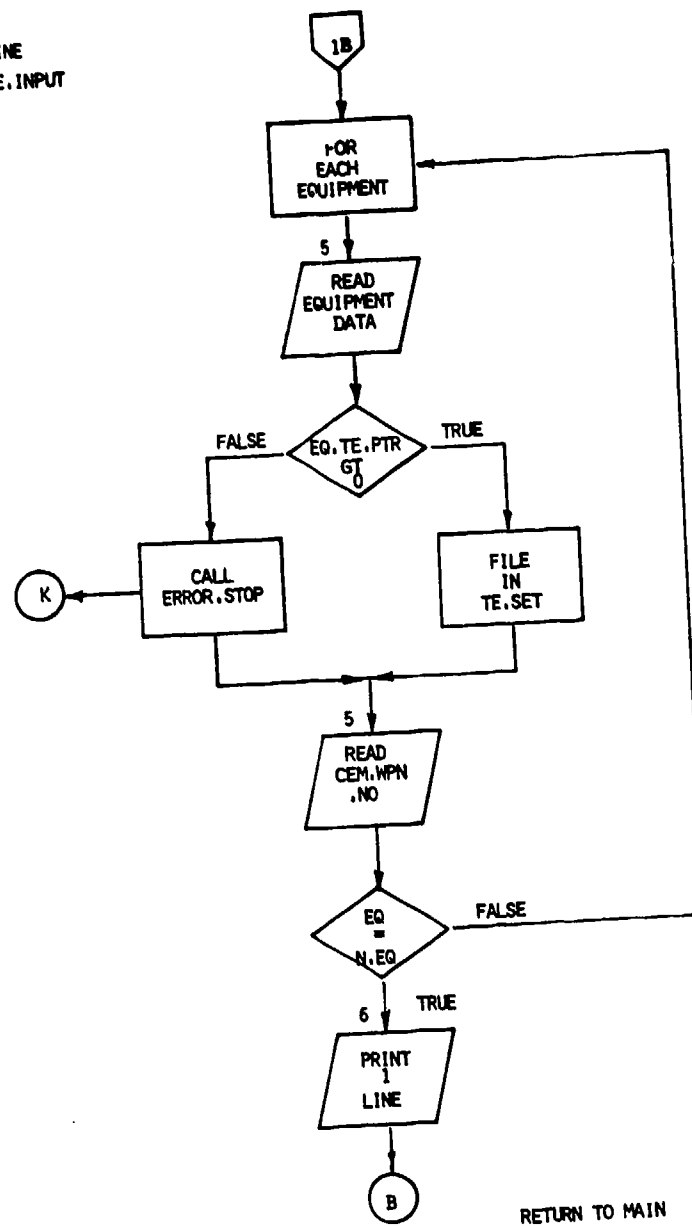


Figure III.2.36 (Cont)

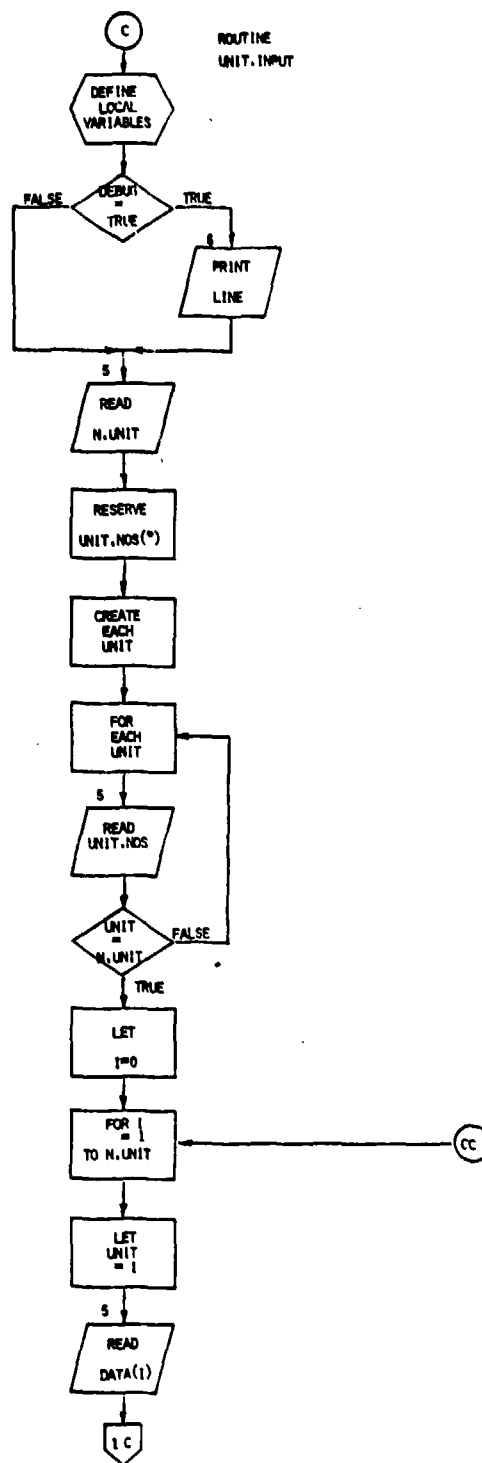


Figure III.2.37

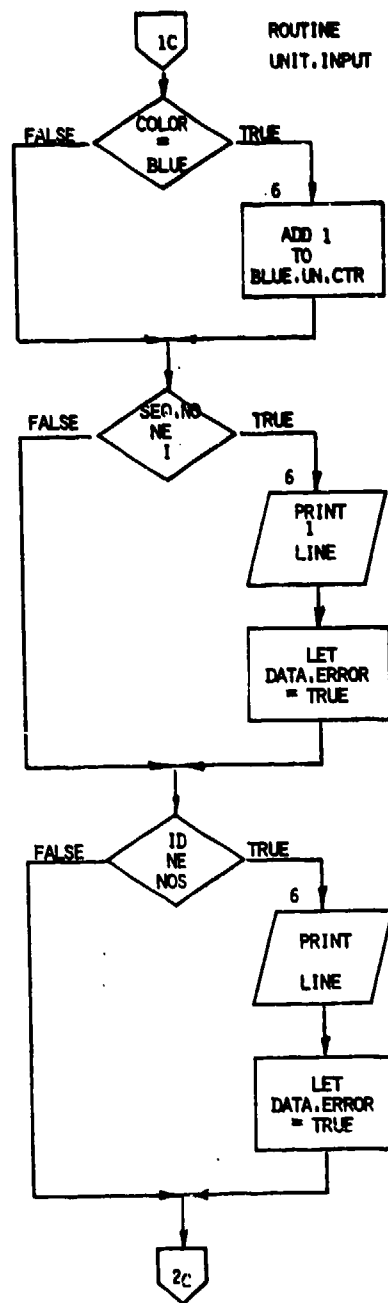


Figure III.2.37 (Cont)

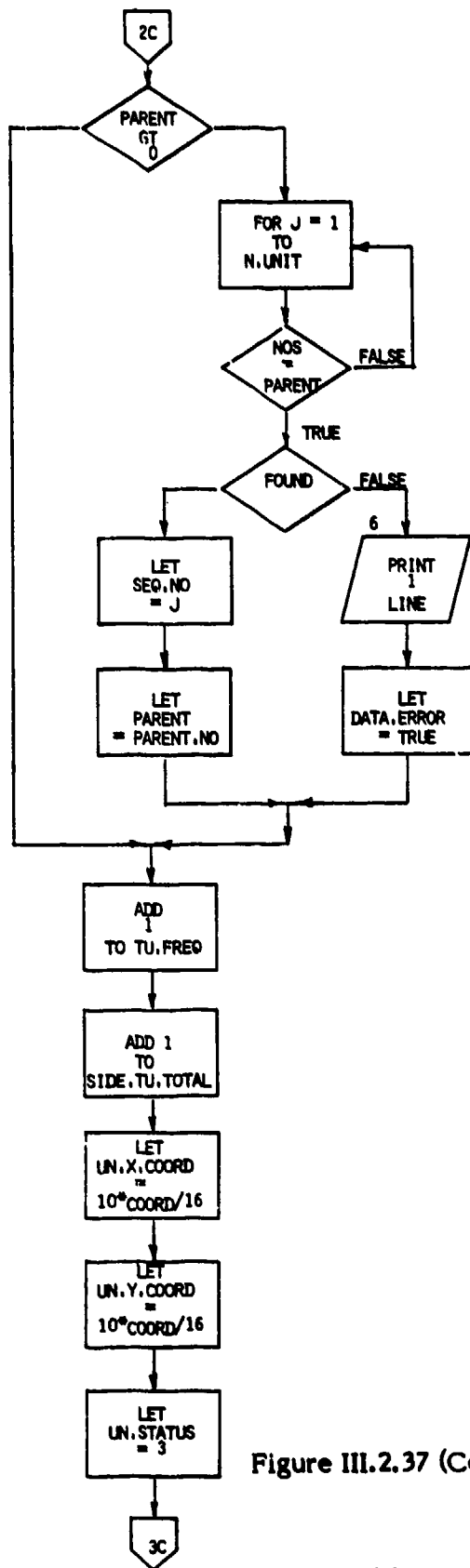


Figure III.2.37 (Cont)

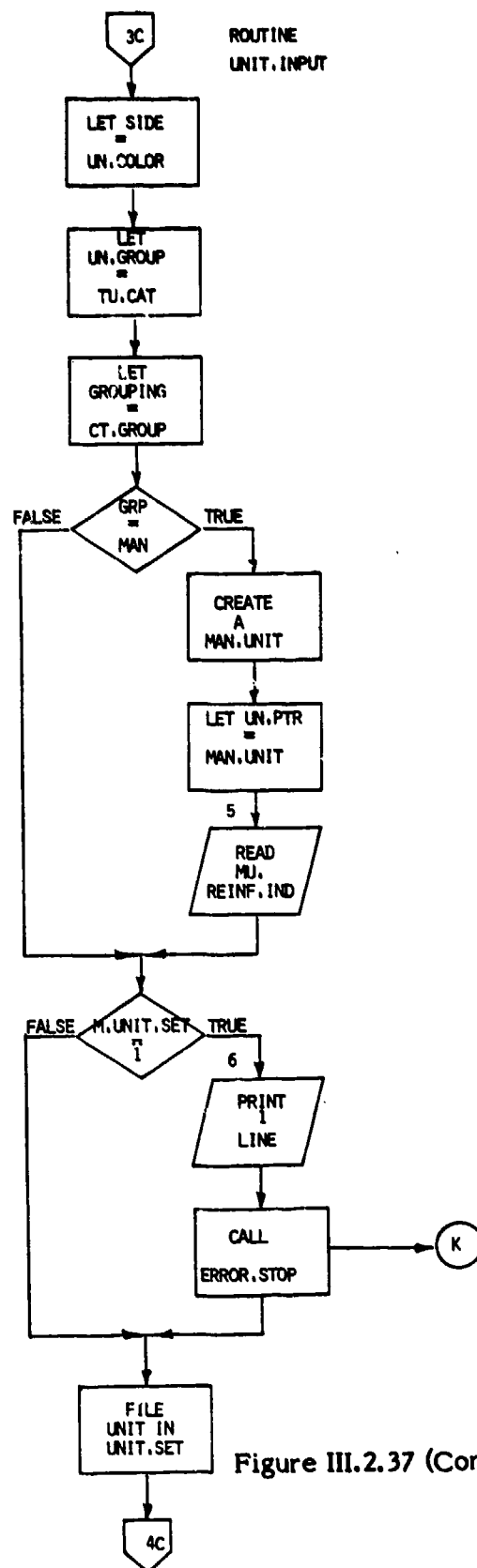


Figure III.2.37 (Cont)

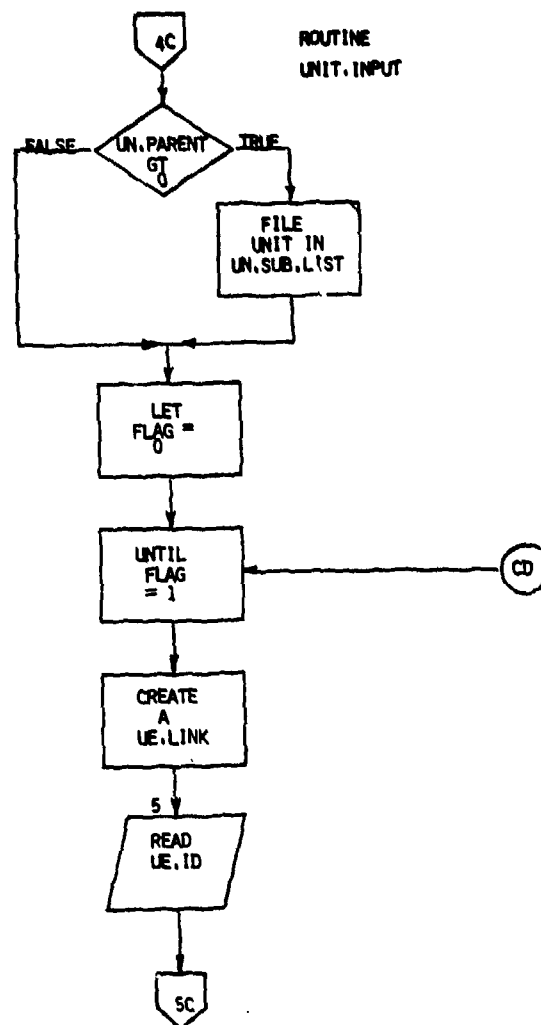


Figure III.2.37 (Cont)

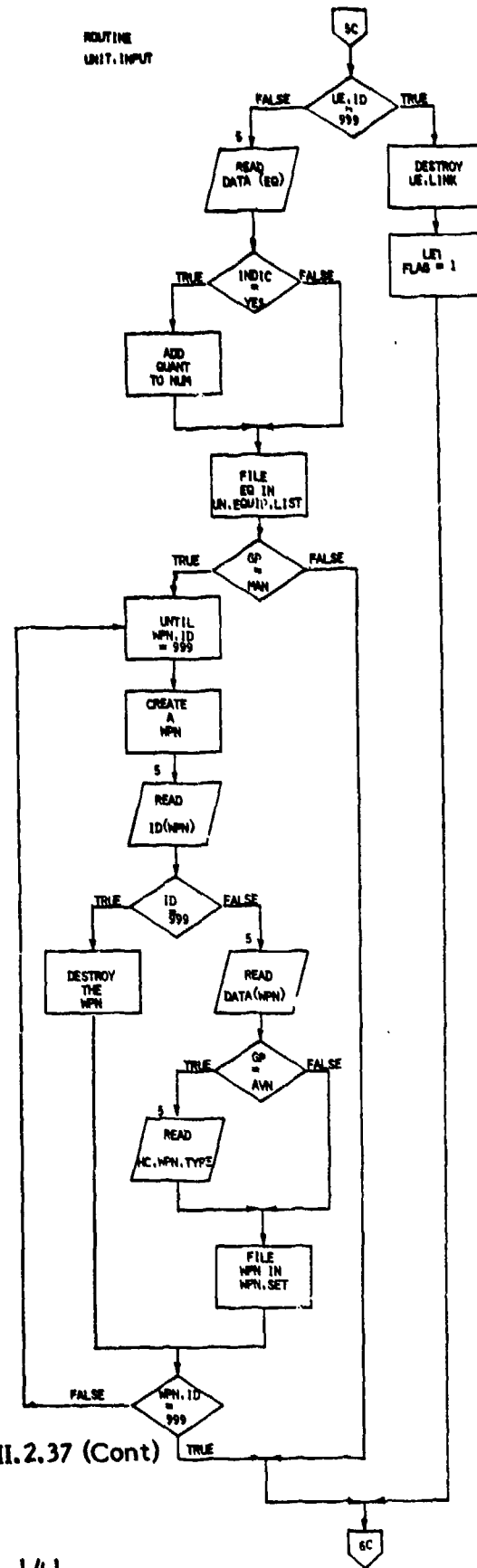


Figure III.2.37 (Cont)

ROUTINE  
UNIT, INPUT

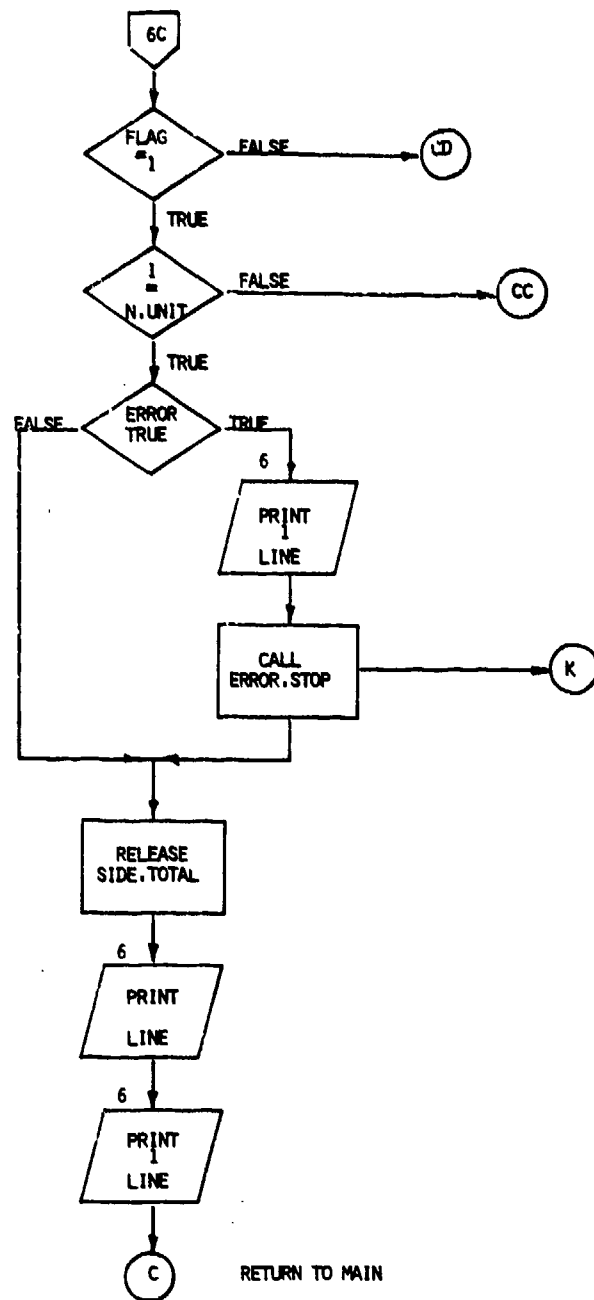


Figure III.2.37 (Cont)

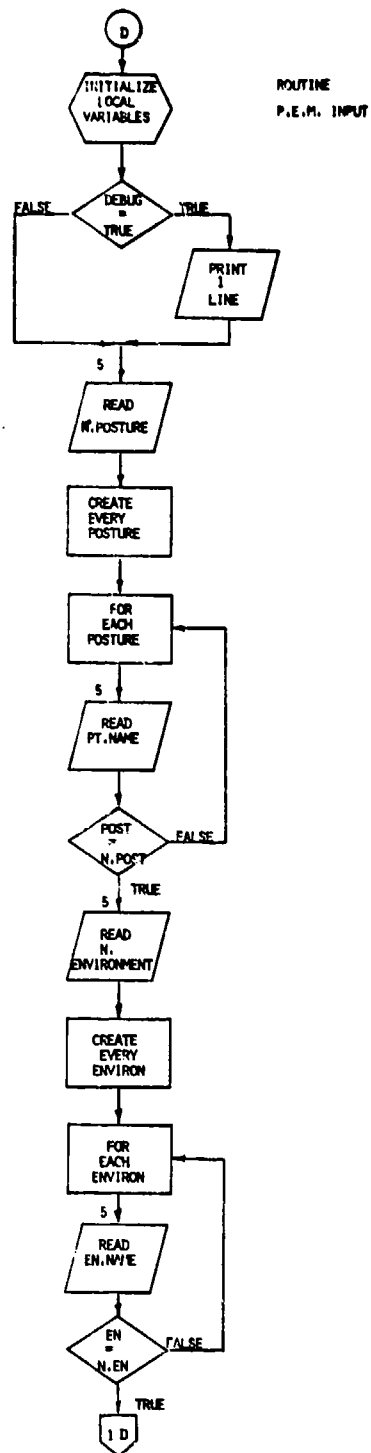


Figure III.2.38

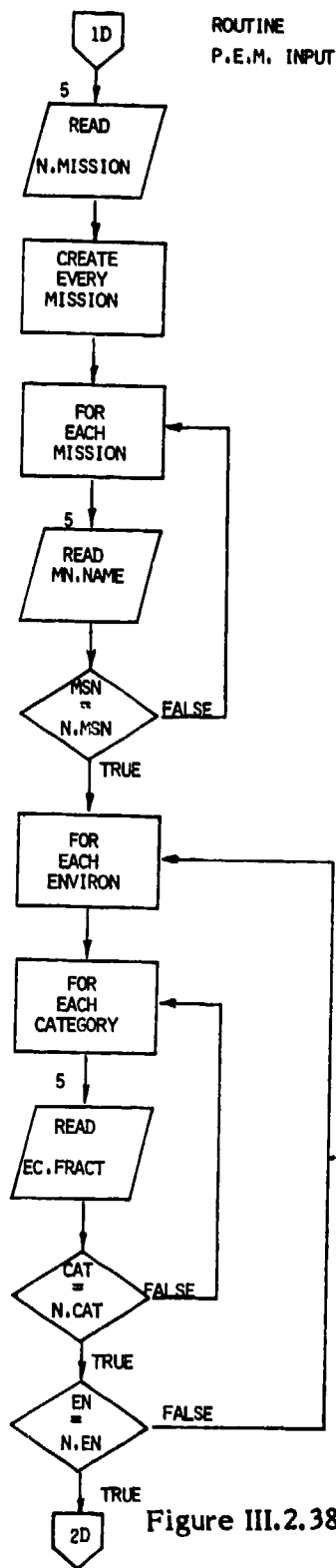


Figure III.2.38 (Cont)

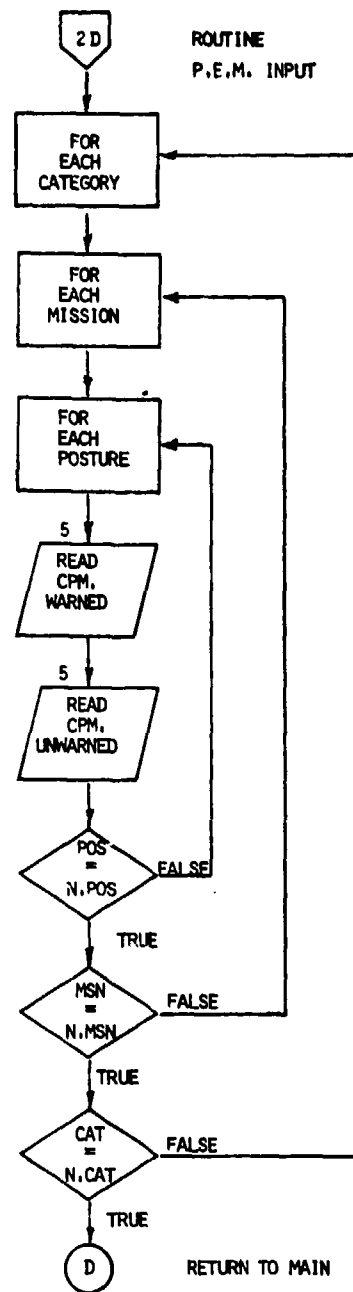


Figure III.2.38 (Cont)

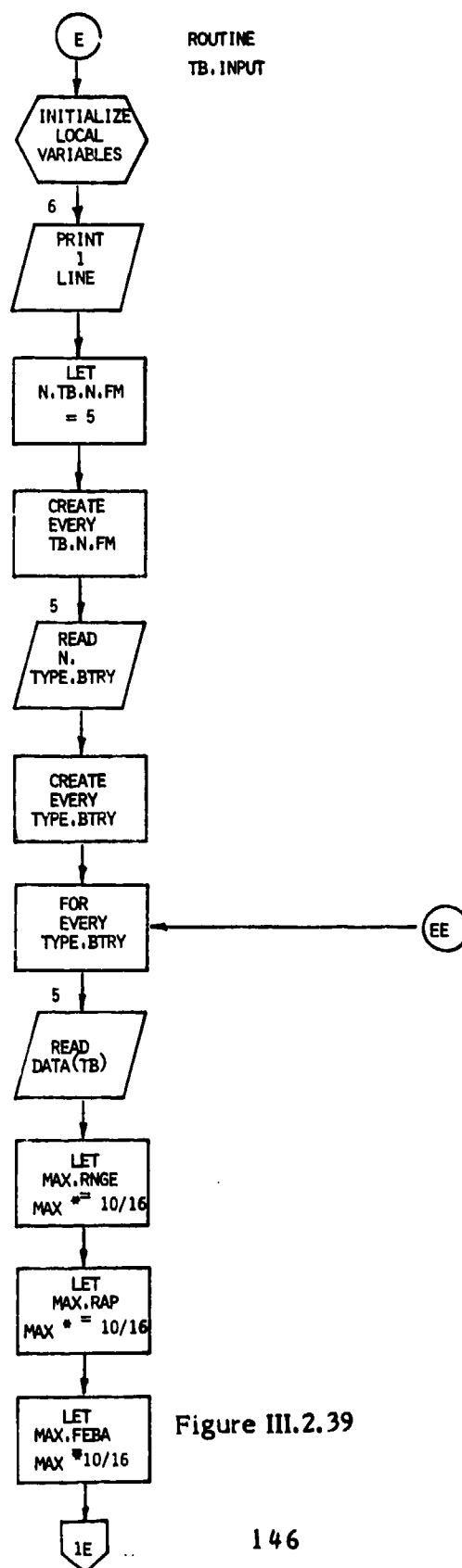
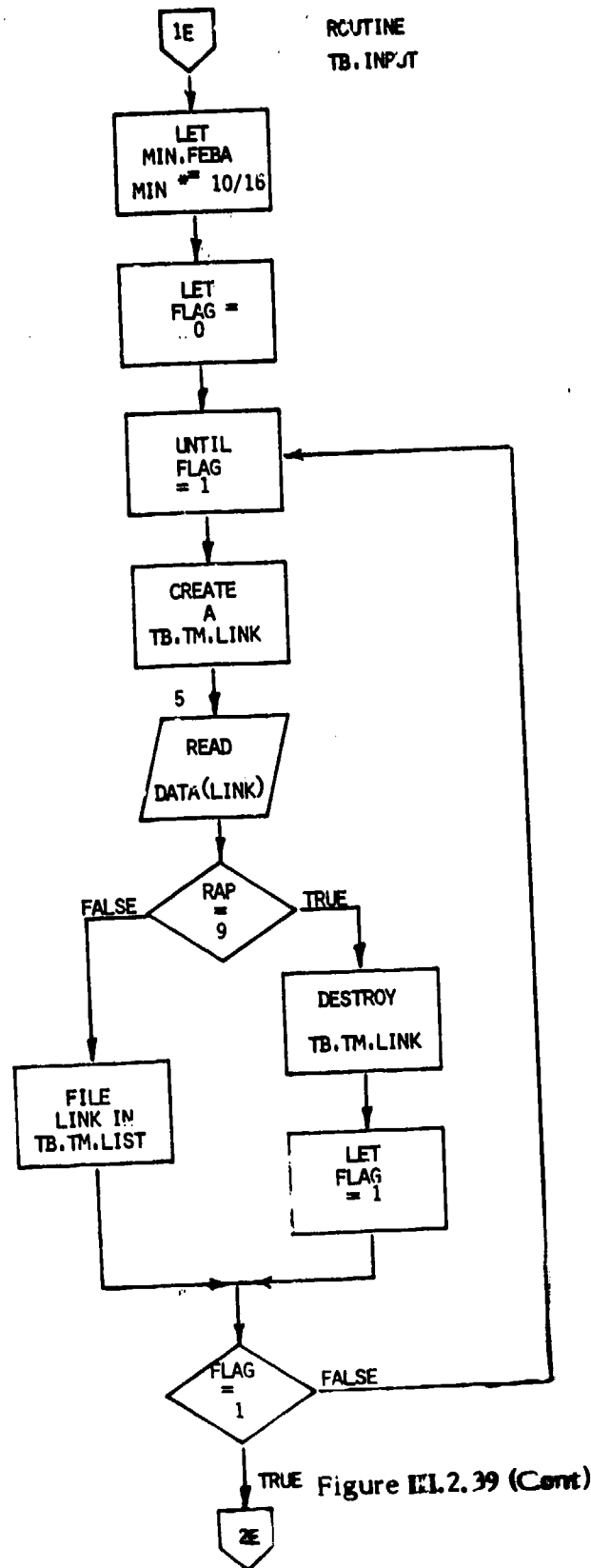


Figure III.2.39



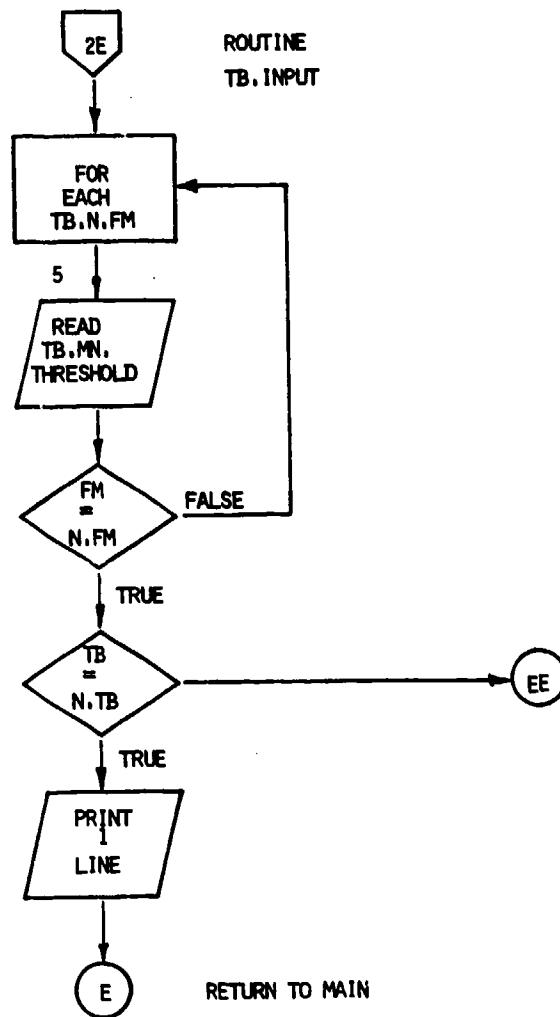


Figure III.2.39 (Cont)

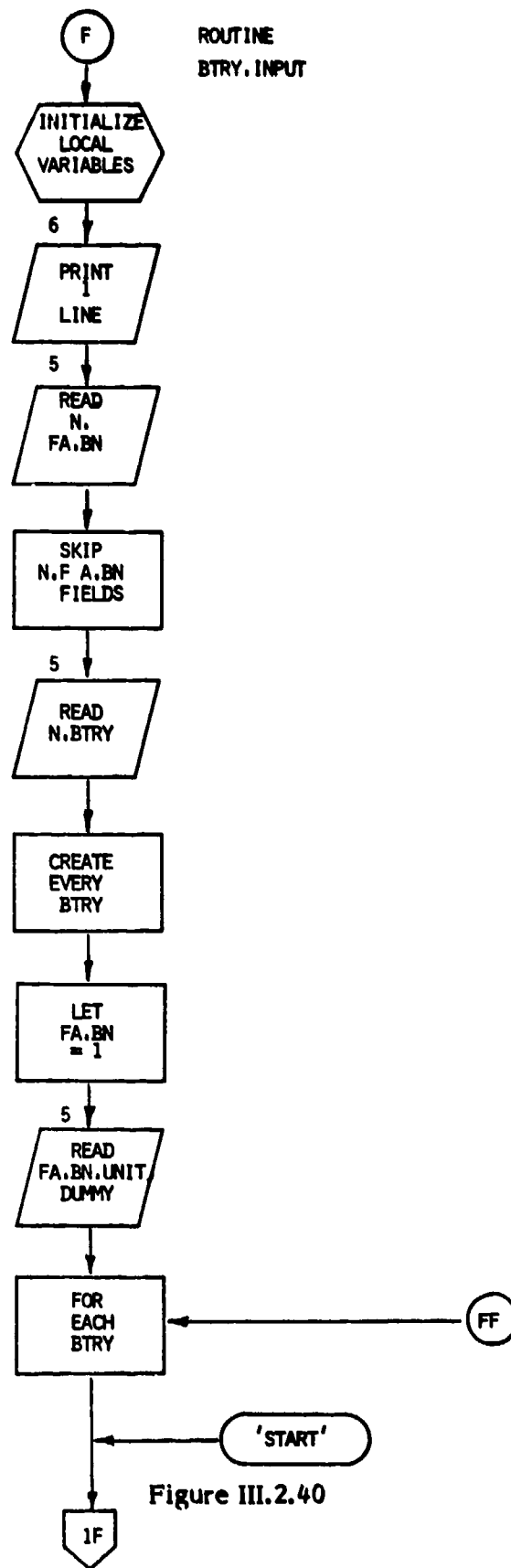


Figure III.2.40

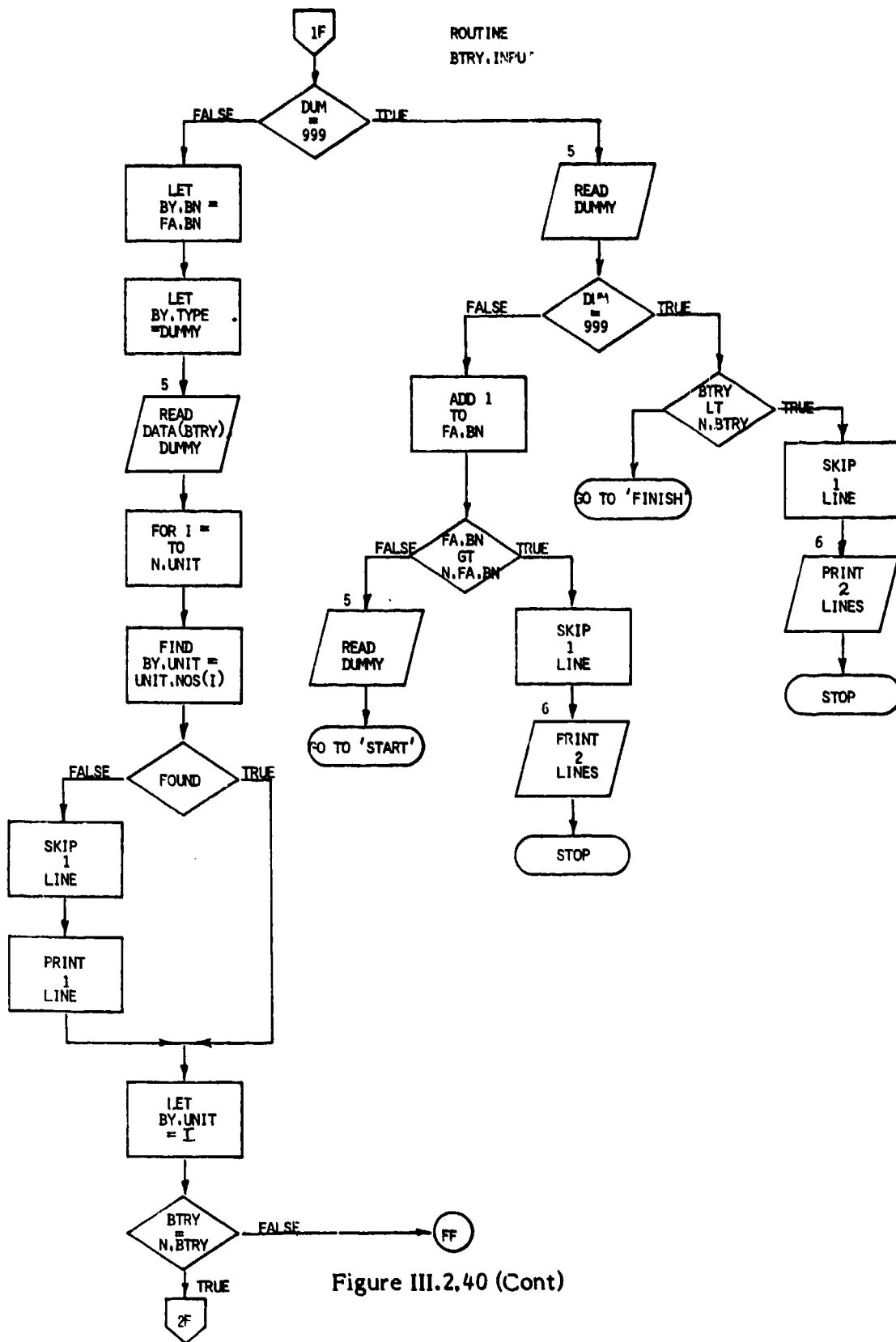
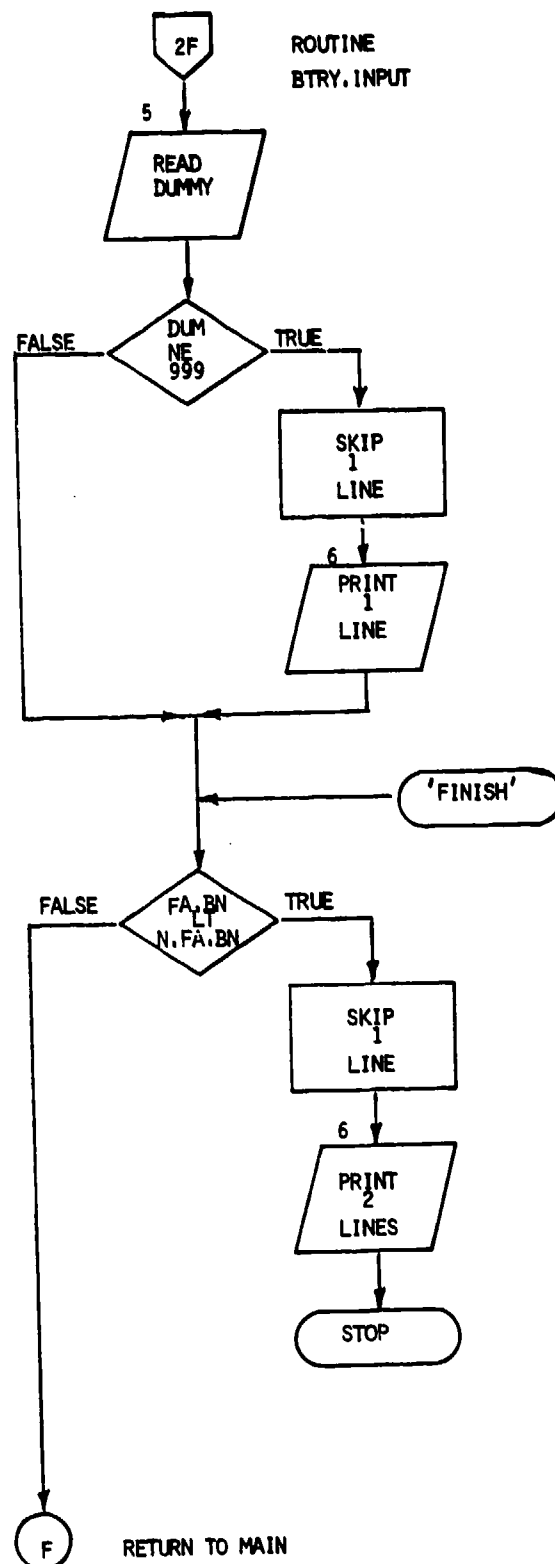


Figure III.2.40 (Cont)



RETURN TO MAIN  
Figure III.2.40 (Cont)

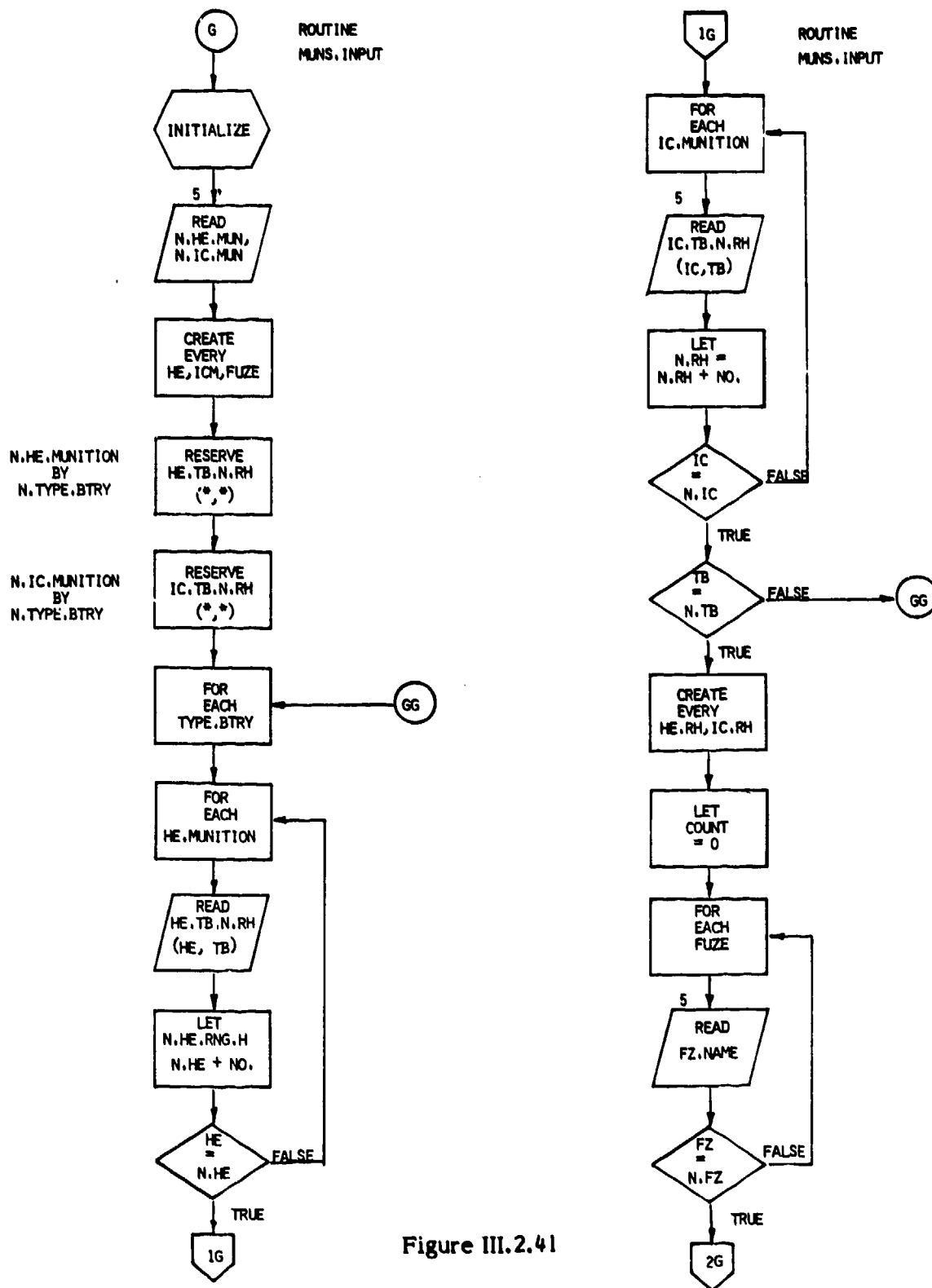


Figure III.2.41

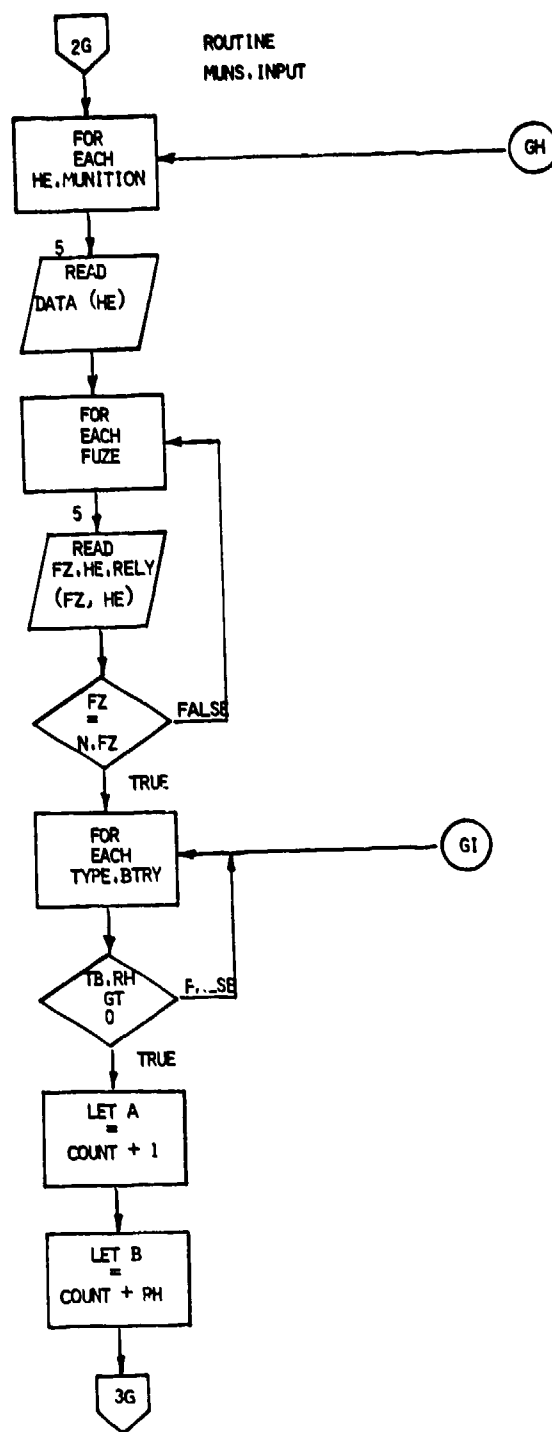


Figure III.2.41 (Cont)

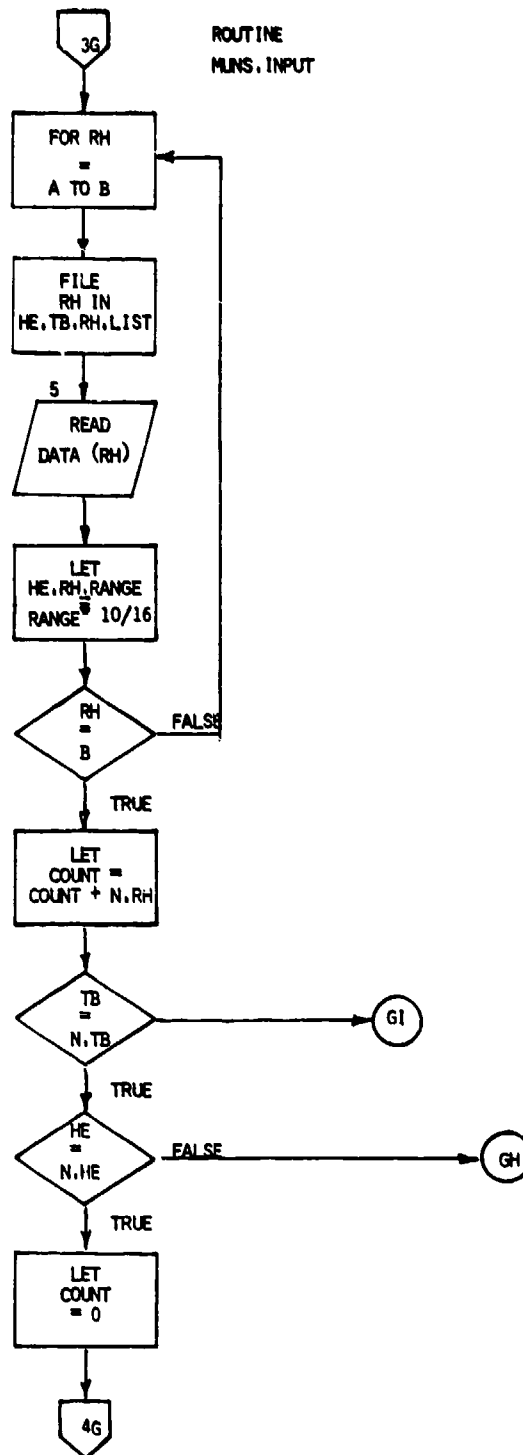
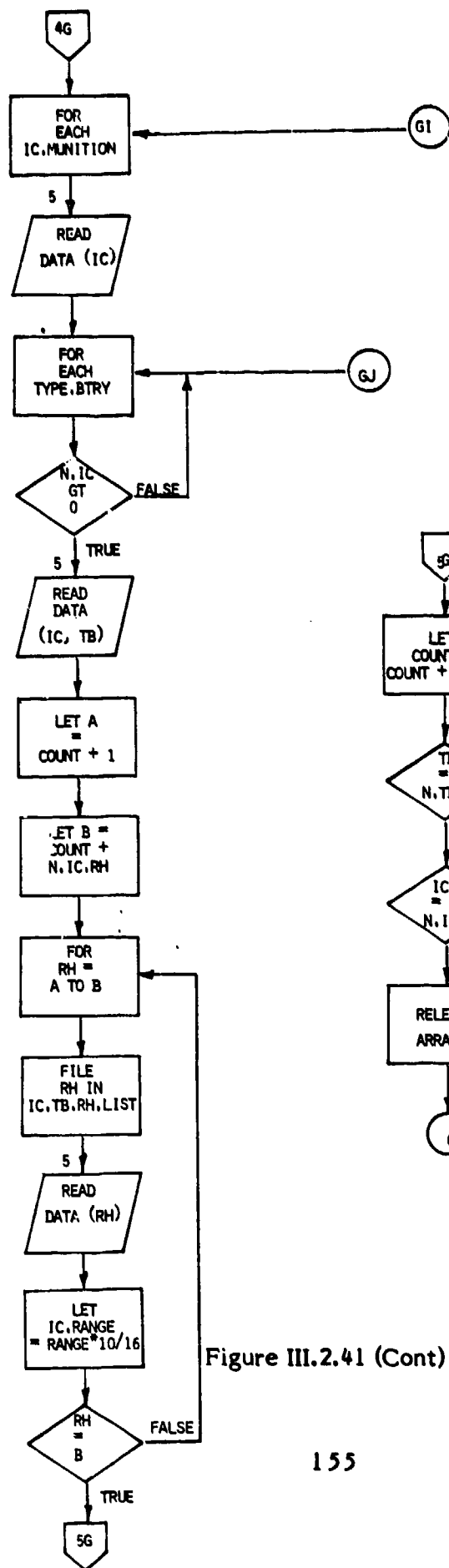


Figure III.2.41 (Cont)

ROUTINE  
MUNS. INPUT



ROUTINE  
MUNS. INPUT

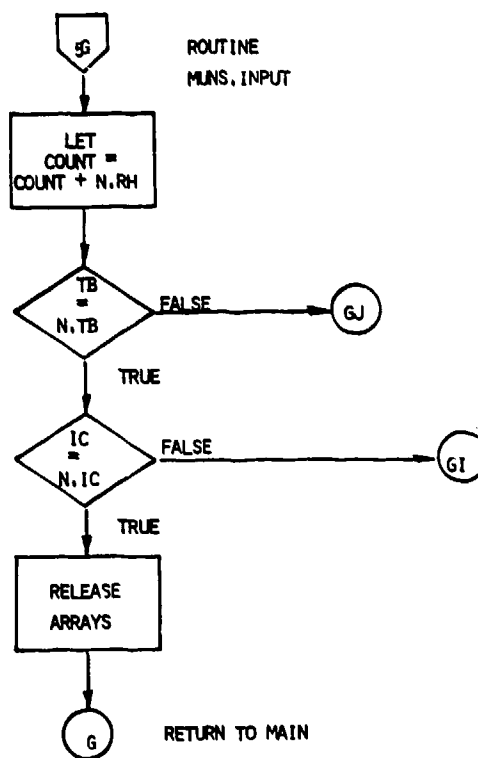


Figure III.2.41 (Cont)

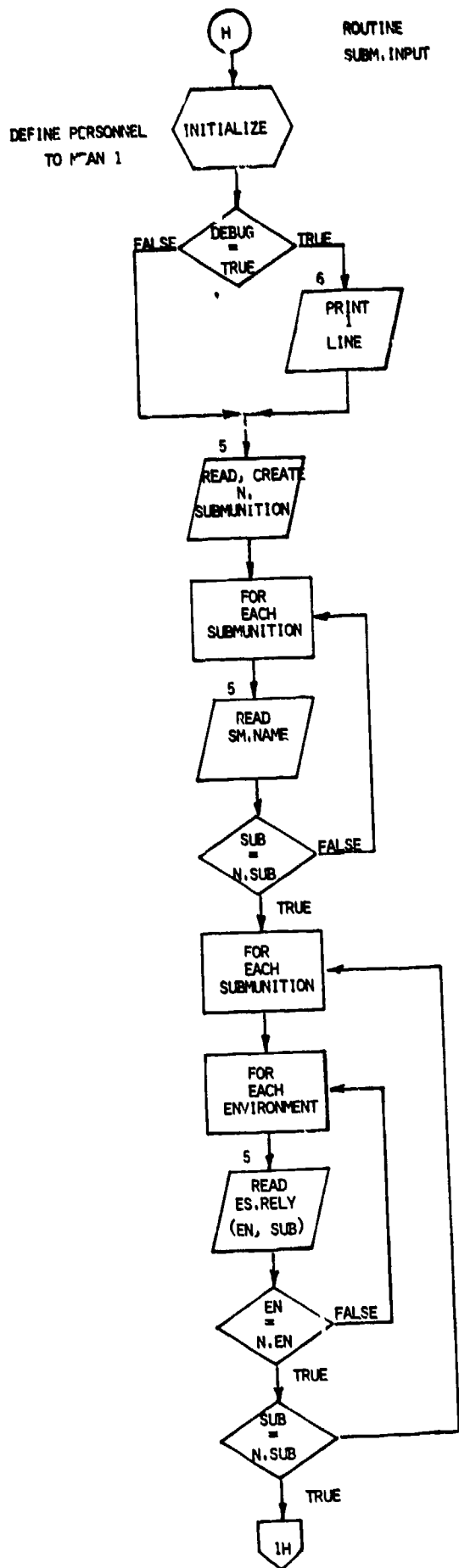


Figure III.2.42

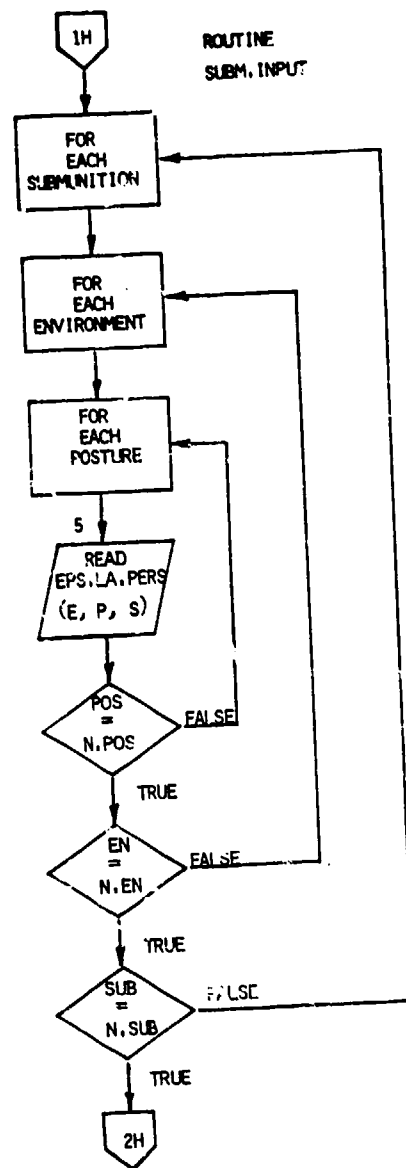


Fig 1.2.42 (Cont)

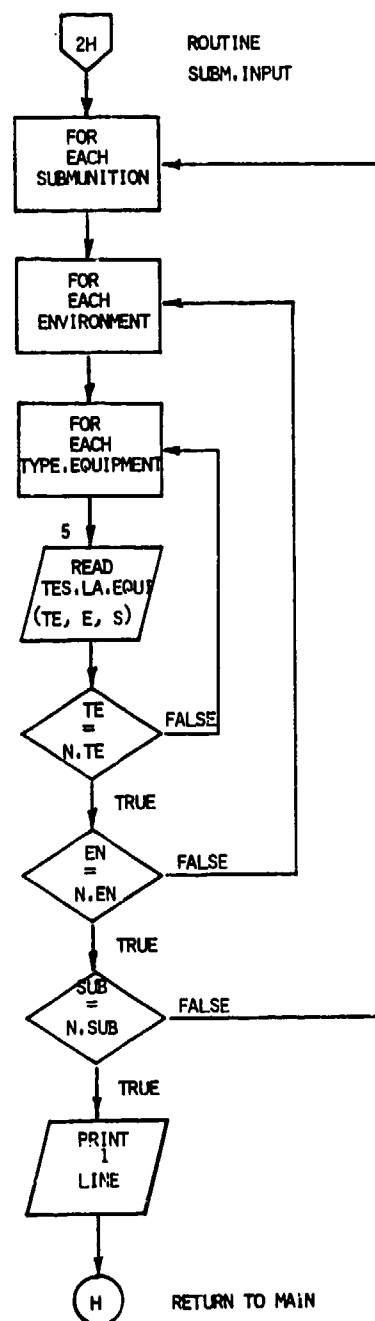


Figure III.2.42 (Cont)

THIS PAGE IS INTENTIONALLY LEFT

BLANK

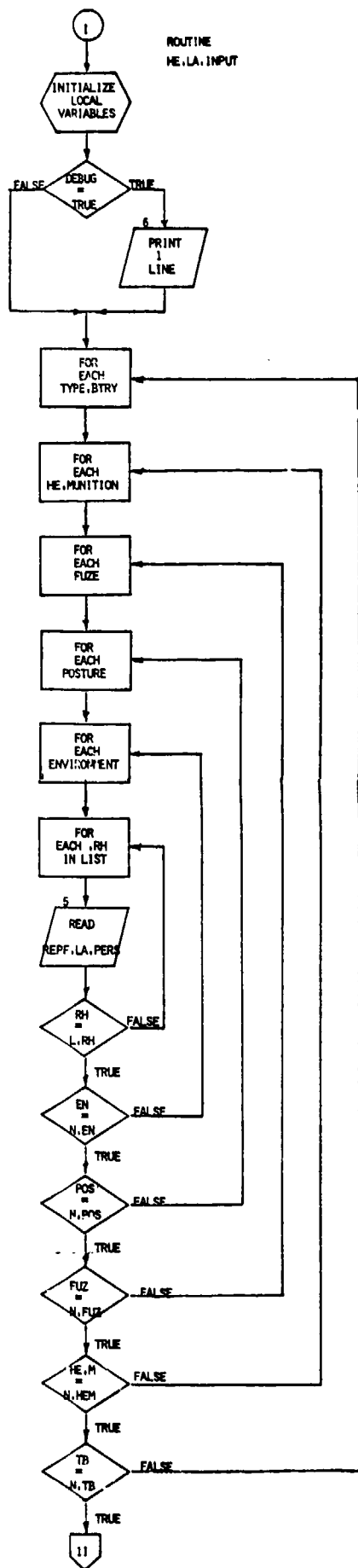


Figure III.2.43

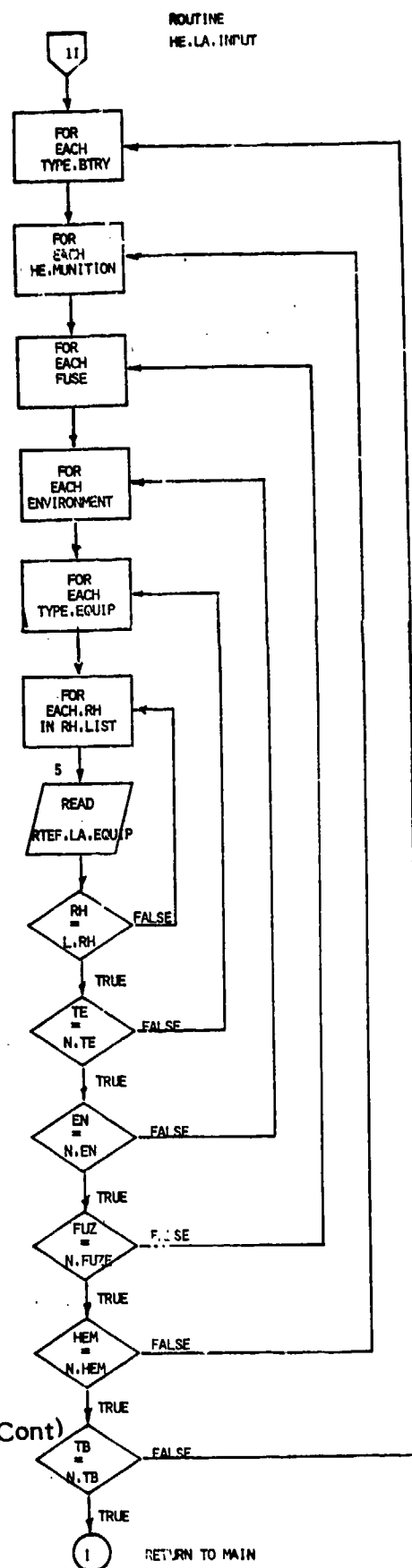


Figure III.2.43 (Cont)

BLUE.UNIT.CNTR  
BY  
23

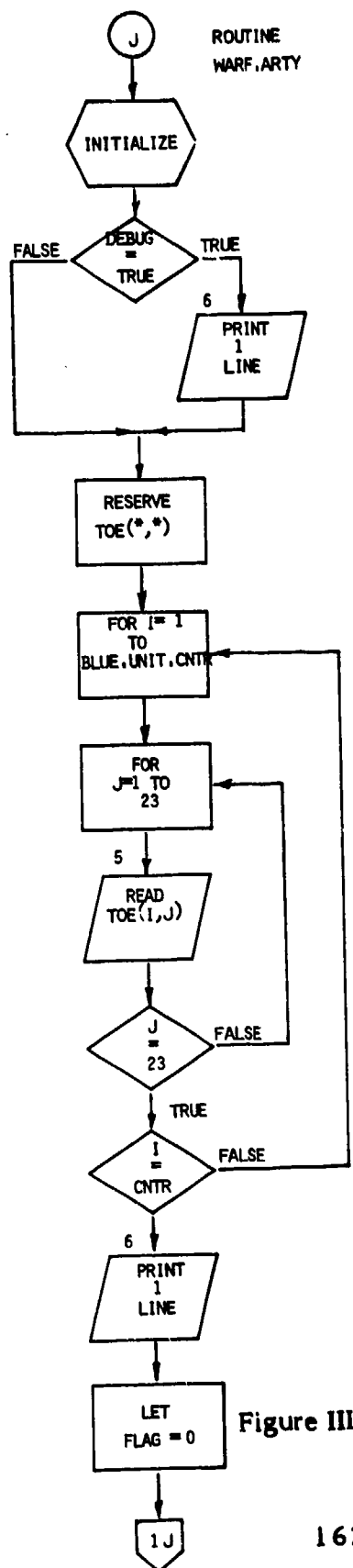


Figure III.2.44

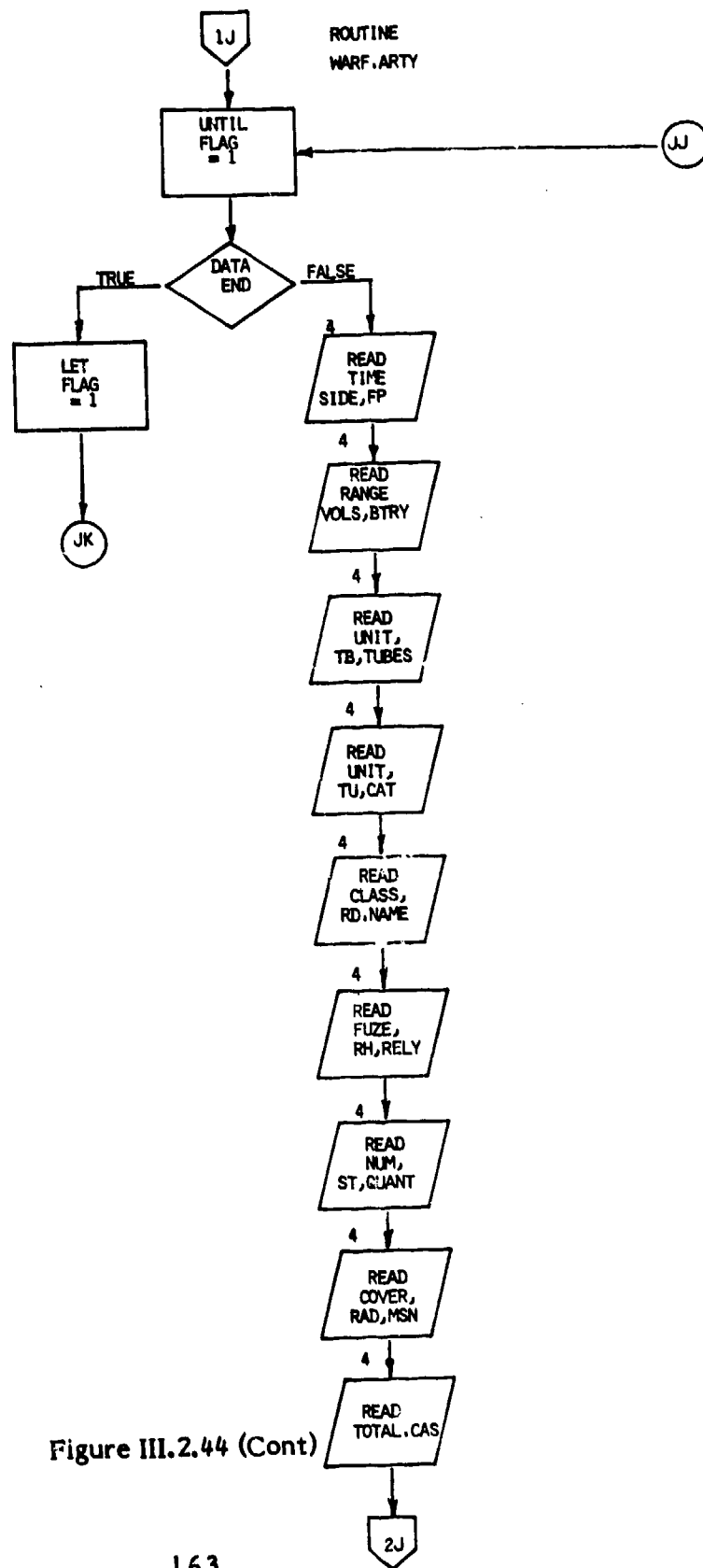


Figure III.2.44 (Cont)

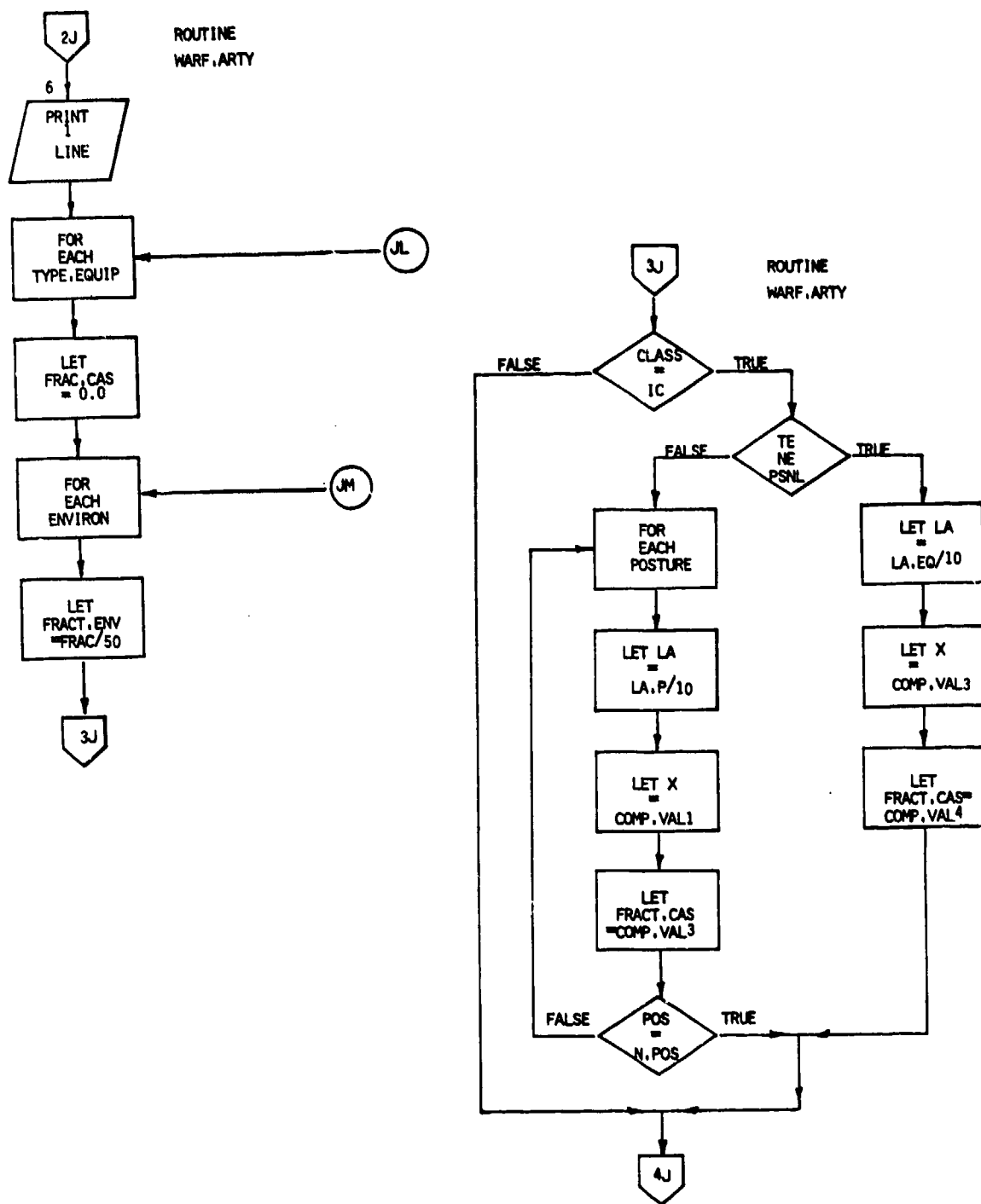
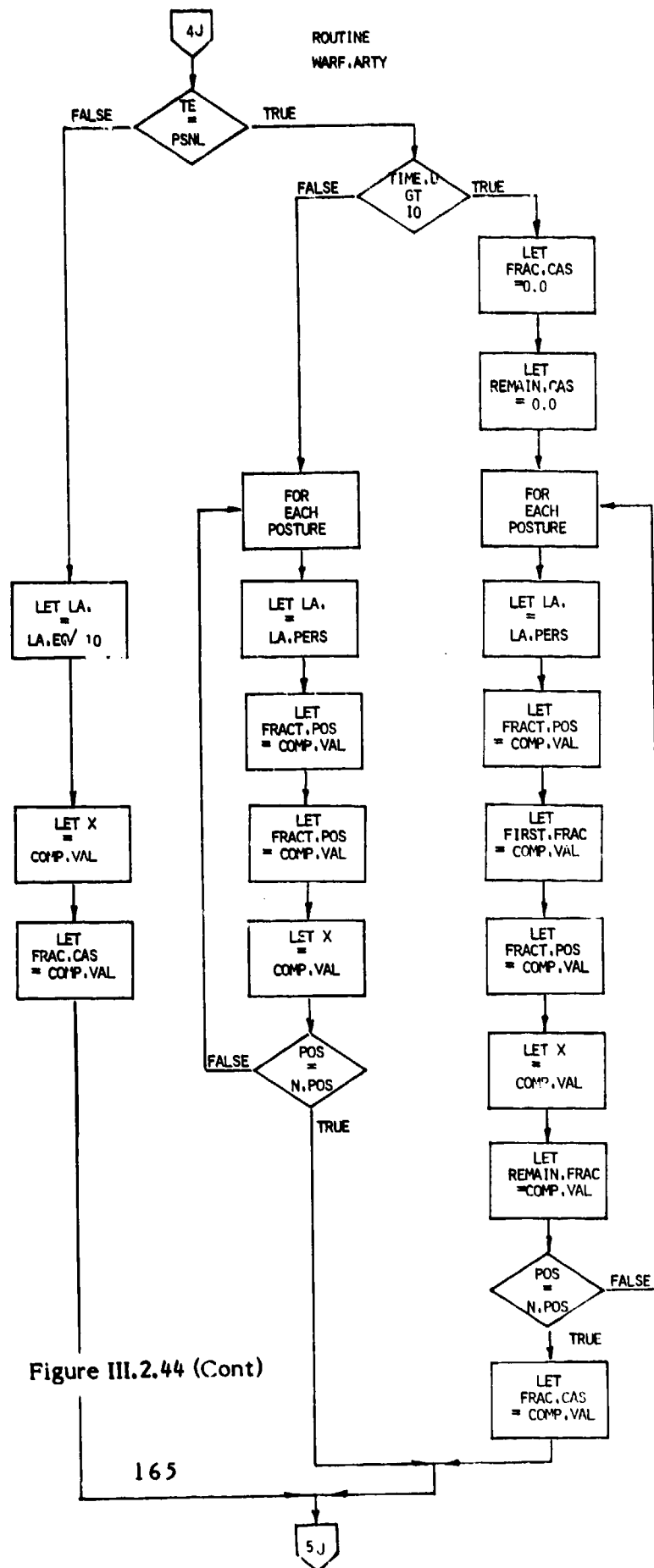


Figure III.2.44 (Cont)



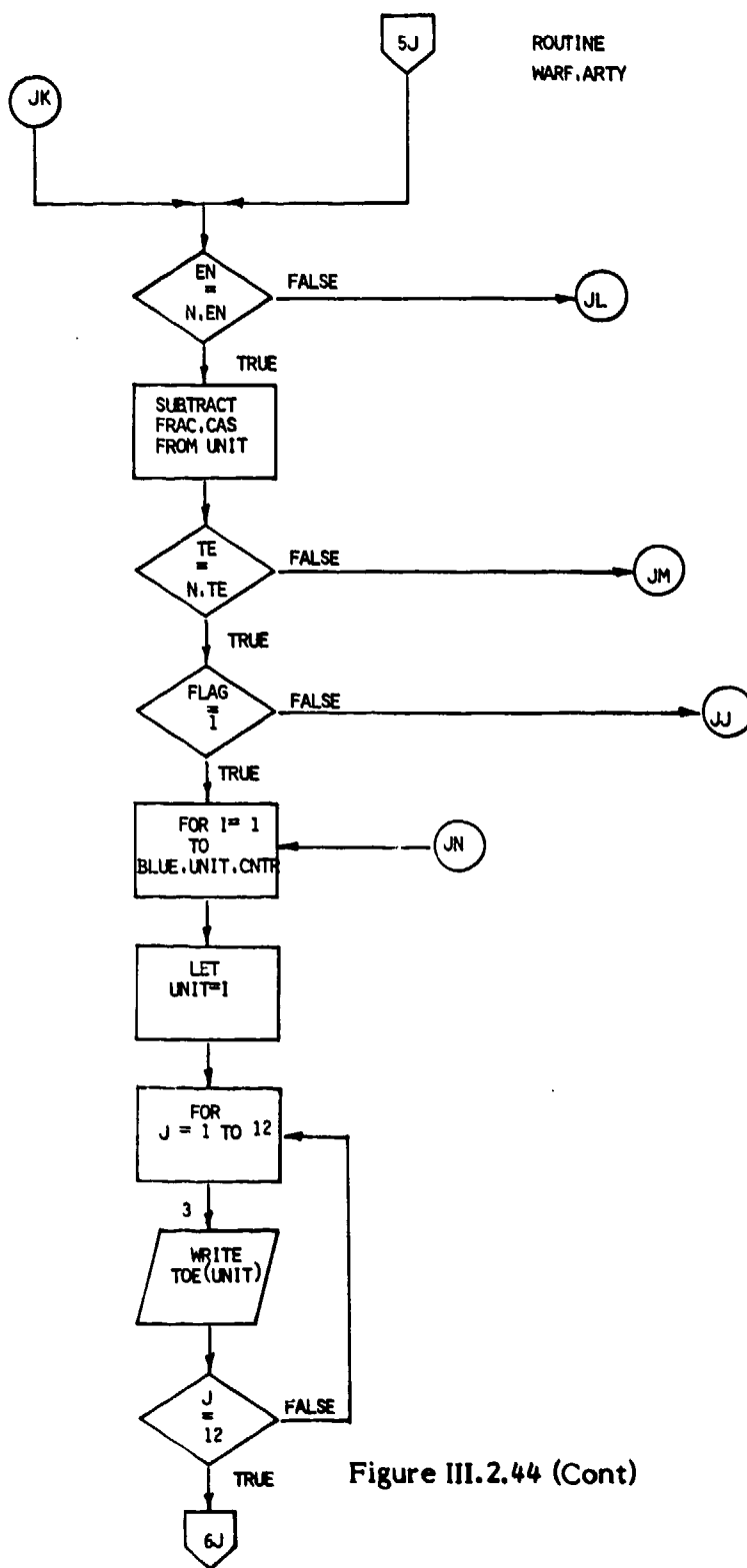


Figure III.2.44 (Cont)

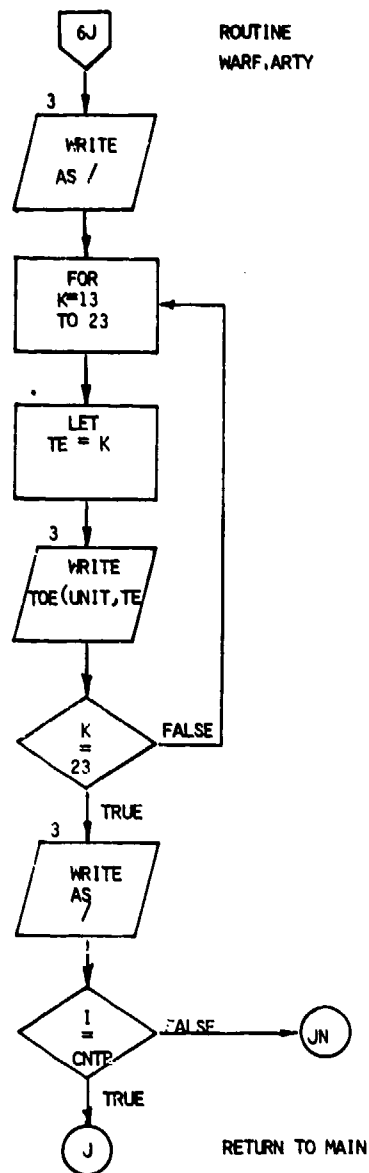


Figure III.2.44 (Cont)

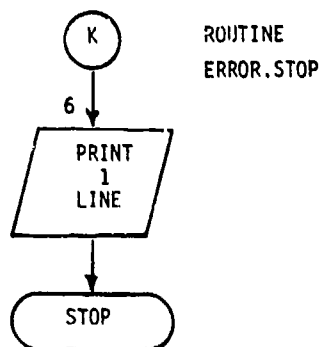


Figure III.2.45

## Chapter 3

### UTILITY WIMP/TOE-IN

**3.1 DESCRIPTION OF PROCESSING:** The WARF Intermediate Materiel Processor (WIMP)/TOE-IN utility program was developed to prepare data (an output) that is used as an input to the WIMP program (Chapter 2). This programs application is contingent on the utilization of the WIMP in the WARRAMP methodology. The program is implemented in the FORTRAN V programming language, and consists of approximately 25 lines of executable computer source code.

**3.1.1 PURPOSE/FUNCTIONS:** The purpose of the WIMP/TOE-IN program is to restate the modeled military units table of organization and equipment (TOE). The military units modeled in the high resolution combat model (COSAGE) have their respective TOE's given in data specifying items of equipment (including personnel) by specific, major equipment item(s). The WARF methodology requires the TOE be stated, or mapped into, groups by artillery (indirect fire) vulnerability category. The present methodology recognizes 22 vulnerability groups. The placement of specific equipment items (by LIN CODE) into one of the artillery vulnerability categories is a function of ballistics analysts and not a function of this program. Additionally, the high resolution combat model does not model all equipment or parts items of a military unit's TOE that realistically would be subject to damage or loss in combat. It is the purpose of this program to account for all equipment items of analytical interest in all of the modeled unit's TOE's. The program achieves this task for the analyst by performing the following functions. First, the input data file element called "RAM/MATRIX" or "WIMP/-MATRIX" (an output from the utility by the same name - Chapter 11) is read sequentially and placed into an array. This is the specific TOE data by type of military unit. Second, the unit data (force array, which is posture unique) is read sequentially and a search of the TOE data (array) is made for a match between the unit's type and the TOE code. When the match is found, the units identification is written to the output file, followed by a copy of the appropriate TOE in formatted write statements.

**3.1.2 PROGRAM INPUT/OUTPUT STRUCTURE:** The WIMP/TOE-IN I/O structure is depicted in figure III.3.1. This general graphic presents the input/output filename.elementname normally associated with the programs operation. The four input data files for the UNITXXX data denote that (at present) the methodology applies four force array (unit) data files for a study analysis. The program is executed once for each of the postures (unit data) portrayed.

**3.1.2.A INPUT DATA AND DATA BASE:** The following input data files are required for the program. The normal, demand-mode runstream for the programs execution requires that this data reside in a cataloged mass-storage (disk resident) file. The data files are required in the order discussed.

- o RAM/MATRIX (or WIMP/MATRIX) data. This input data is produced (as output) from the utility RAM/MATRIX discussed in Chapter 11. The data is formatted with integer and alphanumeric data; a sample of the

file is presented in Figure III.3.2. Twenty three data points are contained in each record; each record defines a TOE for a type of military unit. It has a non-standard record length of 96 columns.

The record format is as follows:

<u>Column</u>	<u>Descriptor</u>	<u>Field</u>
1	Blank	1X
2-3	Alpha TOE identifier	2A
4-5	Integer TOE sequence number	2I
6-94	Integer equipment quantities for each of the (22) vulnerability groups	22(14)

- o Unit (force array) data. This file is a copy of the manually or SUSF created data file for the high resolution ground combat model (COSAGE). Originally produced as a free-formatted, all integer mode data file, this program requires that the first record for each unit have the unit identification and type unit identification in the format specified below. A sample of the data file is depicted in figure III.3.3. In the example shown the first record that contains data to be used by the program would be record 41. The data record format is as follows:

<u>Column</u>	<u>Description</u>	<u>Field</u>
1-6	Skipped, blanks and user sequence number, or total units (record 1)	6X
7-9	Unit identification number	3I
10-12	Skipped, blanks	3X
13-14	Type unit identification	2I
15-80	Not considered, advance to next record	--

3.1.2.B Output Data and Data Files: This program produces one (1) output file, which contains the restated TOE's. The file element name is generally the same as the program, with the posture of the input file appended as: TOE-IN-XX; the "XX" is replaced through editing of the procedure of runstream file to provide an audit trail of data. A sample output file is depicted in Figure III.3.4. The output

data produced is positional and sequentially produced; there are no labels identifying the type unit TOE. Thus the first unit type is in the force array (unit data). Two records in decimal format are produced for each TOE with a blank record between TOE's. Twenty three data values are written, the first position always accounting for personnel, and because personnel are not of analytical interest in the WARF methodology, is manually set to zero. The record formats are as follows:

<u>Column</u>	<u>Descriptor</u>	<u>Field</u>
Record 1:		
1-96	Personnel their vulnerability group 11 equipment quantity.	12(1X,F7.2)
Record 2		
1-88	Vulnerability group 12 thru 22 equipment quantity.	11(1X,F7.2)
Record 3:		
1-80	Skipped record.	--

3.1.2.C Data Element Dictionary: The data elements used in the program are as follows:

<u>Variable Name</u>	<u>Mode</u>	<u>Value</u>	<u>Definition</u>
NCOCED1(I)	Integer	0-500	One-dimensional array for the type unit value from the RAM/MATRIX data file.
NCODE2	Integer	1	Program run variable reset at input for the type unit from the UNITXXX data file
NQTY(I,J)	Integer	0-500, 0-22	Two-dimensional array for the TOE of each type unit from the RAM-MATRIX data file.

PERS	Real	000	Constant program variable for the personnel quantity value.
QTY (I,J)	Integer	0-500, 0-22	Two-dimensional array to hold the TOE data for the output file.
NTYPE	Integer	I	Program run variable re-set at input for the unit identification number of the unit record being read.

3.1.3 PROGRAM PROCESSING: The program processes are straight forward as: read data, logic tests for equality and write output data. The program consists of one "routine" and thus common blocks and linkages are not used.

3.1.3.A PROGRAM RUN DESCRIPTION: The program runstream is depicted in Figure III.3.5. The program's short length enable user's to run the program in the demand mode from the terminal. See Volume I, Section III, Chapter 3 of this document set for additional details.

3.1.3.B PROGRAM LOGIC: The programs logic is graphically presented in Figure III.3.6.

3.1.3.C PROCESSING FEATURES: The processing features are straight forward logical tests of the data and can be determined from a review of the program listing, Figure III.3.7, and logic diagram, Figure III.3.5.

3.2 OPERATING ENVIRONMENT: The WIMP/TOE-IN is compiled on and operated on the USACAA UNIVAC 1100/82 operating system which is a secure system in compliance with Department of the Army security directives.

3.2.1 Hardware: The only explicit hardware requirements is that provided by the UNIVAC 1100/82 OS and a compatible terminal (UTS 400). The operating system dynamically handles on-line storage requirements for the input/output files. Object code and permanent files for the I/O functions and runstream must be explicitly cataloged (assigned) to disk-pack file(s) either removable or fixed. They may reside on a private disk or system (public) disks.

3.2.2 SUPPORT SYSTEM: This program requires the following for demand processing. The program has no Executive-8 release dependent characteristics.

3.2.2.A SYSTEM SOFTWARE: The program utilizes the following system processors.

- o ED - text editor processor
- o MAP - relocatable code collector

3.2.2.B COMPILER: The operating system library contains the Fortran V compiler for program compilation. The compiler is invoked by the @FTN command; refer to UP 40GO for diagnostic, and supporting work file (temporary) requirements.

3.2.3 DATA BASE: The data base files necessary for program operation were discussed in paragraph 3.1.2.A. These files exist as user cataloged file elements and are not part of a formal data base structure.

3.3 MAINTENANCE PROCEDURES: The relative size of this program minimizes the maintenance on the program. The program is well documented with comment lines. Modifications to the program must be clearly denoted by programmers in the manner shown in the listing, Figure III.3.6. Page by page changes to this manual must be made when changes to the program are made which alter the logic, inputs or outputs. The following information is pertinent to the program's maintenance (space provided for notes):

Source Code Filename.Elementname:

Absolute Code Filename.Elementname:

Space required, source code:

Space required, Object (absolute) code:

Archived tape location:

Read/Write protect keys: Established by the current custodian.

3.3.1 PROGRAMMING CONVENTIONS Each function of the program is documented. Line labels begin in column 1 and source code in column 7; standard Fortran techniques are employed with FORTRAN V conventions.

3.3.2 VERIFICATION PROCEDURES: The program is verified by test executions (runs) of the program followed by hand calculations and comparisons of the results.

3.3.3 ERROR CORRECTION PROCEDURES: Debugging and error correction is accomplished by an examination of the compiler listing and examination of the run execution (PRINT\$) file.

3.3.4 SPECIAL MAINTENANCE PROCEDURES: None.

WIMP/TOE-IN STRUCTURE

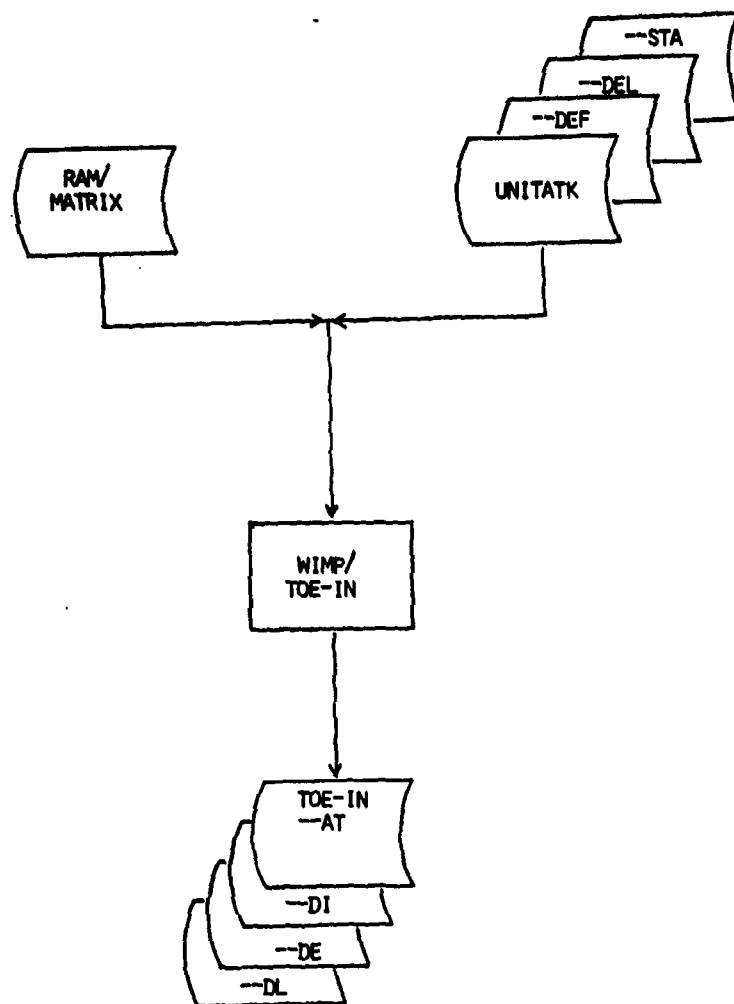


Figure III.3.1



UNCLASSIFIED\*\*\*EXAMPLE OF WIMP INPUT FILE UNT\*\*\*UNCLASSIFIED

```

1: 277          001 002 003 005 006 007 009 010 011 012
2:             013 015 056 016 017 018 020 021 022 023
3:             025 026 027 028 030 037 031 032 033 035
4:             036 037 038 040 041 042 043 045 058 046
5:             047 048 050 051 052 053 055 059 060
6:
7:             061 062 063 065 066 067 068 070 071 072
8:             073 075 116 076 077 078 080 081 082 083
9:             085 086 087 088 090 117 091 092 093 095
10:            096 097 098 100 101 102 103 105 118 106
11:            107 108 110 111 112 113 115 119 120 185
12:            190 195
13:
14:            121 122 123 125 126 127 128 130 131 132
15:            133 135 176 136 137 138 140 141 142 143
16:            145 146 147 148 150 177 151 152 153 155
17:            156 157 158 160 161 162 163 165 178 166
18:            167 168 170 171 172 173 175 179 180
19:
20:            201 202 203 205 206 207 208 210 211 212
21:            213 215 216 217 218 220 221 222 223 225
22:            226 227 228 230 231 232 233 235 236 237
23:            238 239 240 241 242 243 245 246 200
24:
25:
26:            501 502 503 505 506 507 508 510 511 512
27:            513 515 516 517 509 518 519 520 591 500
28:
29:            601 602 603 605 606 607 608 610 611 612
30:            613 614 615 609 691 600
31:
32:            701 702 703 705 706 707 708 710 711 712
33:            713 715 791 700
34:
35:            801 802 803 804 805 806 807 808 809 810
36:            811 812 813 814 815 816 817 820 821 822
37:            823 826 827 830 831 832 835 836 837 838
38:            839 840 841 842 845 897 898 899
39:
40:
41: 001 001 11 3900 3920 005 2 150 2
42:            03 005 1 005 1 1 006 1 1 007 1 1 999
43:            09 001 2 020 1 1 999
44:            14 002 2 026 1 1 999
45:
46: 007 002 05 3900 3780 005 2 150 2
47:            01 036 2 001 1 1 999
48:            07 004 1 017 1 1 999
49:            07 001 2 020 1 1 999
50:            14 001 2 026 1 1 999
51:            26 001 2 001 1 1 999
52:
53: 003 003 11 3900 3740 005 2 150 2
54:            03 005 1 005 1 1 006 1 1 007 1 1 999
55:            09 001 2 020 1 1 999
56:            14 002 2 026 1 1 999
57:

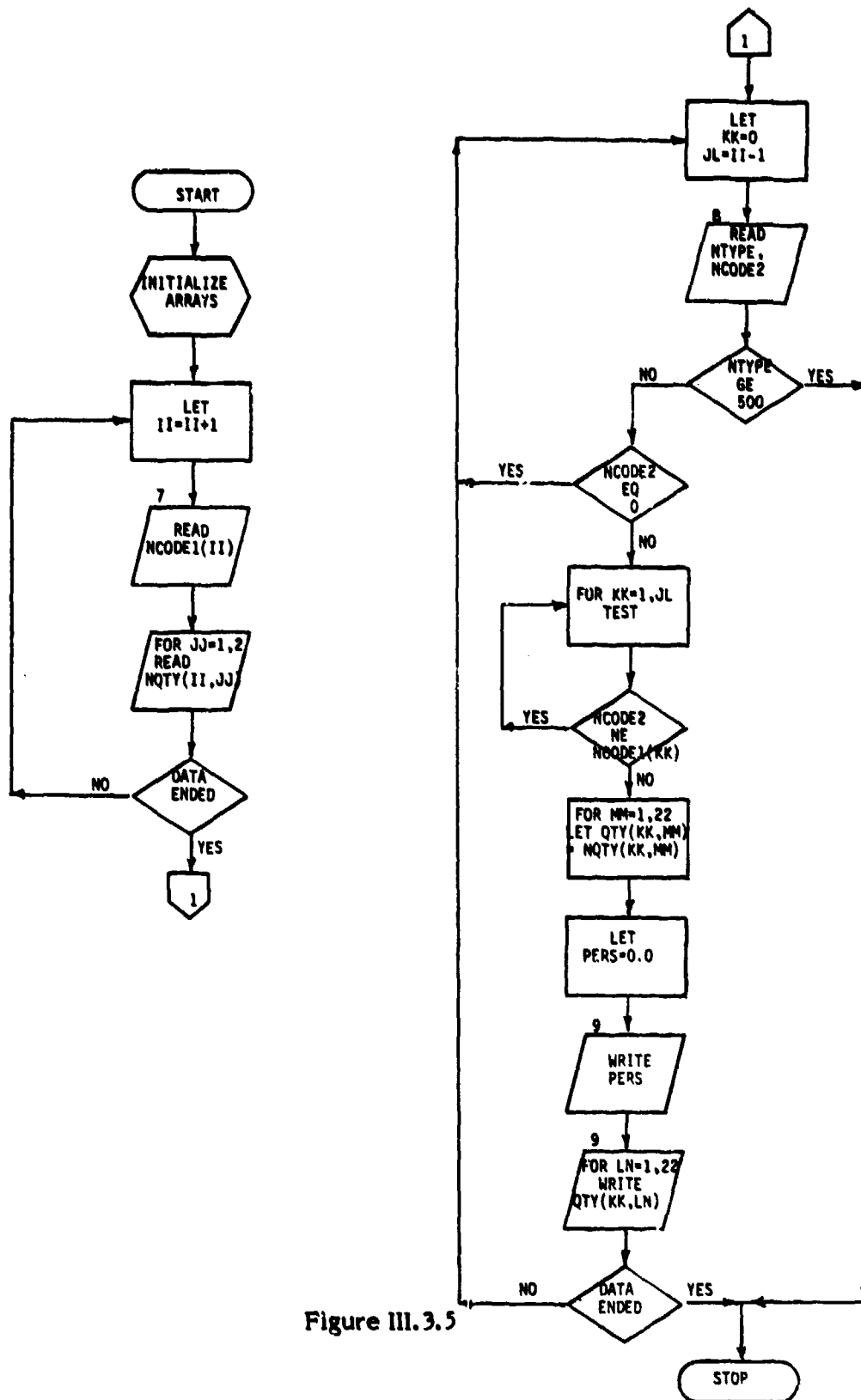
```

Figure III.3.3

UNCLASSIFIED\*\*\*EXAMPLE OF OUTPUT DATA OF UTILITY WIMP/TOE-IN\*\*\*UNCLASSIFIED

1:	.00	.00	.00	.00	1.00	.00	.00	.00	.00	.00	.00	.00
2:	.00	.00	107.00	19.00	84.00	68.00	4.00	26.00	.00	.00	.00	.00
3:												
4:	.00	.00	1.00	.00	1.00	.00	.00	1.00	.00	.00	.00	.00
5:	.00	.00	68.00	11.00	30.00	28.00	.00	2.00	.00	.00	.00	.00
6:												
7:	.00	.00	.00	.00	1.00	.00	.00	.00	.00	.00	.00	.00
8:	.00	.00	107.00	19.00	84.00	68.00	4.00	26.00	.00	.00	.00	.00
9:												
10:	.00	.00	.00	.00	5.00	2.00	.00	4.00	.00	.00	1.00	.00
11:	.00	.00	128.00	1.00	40.00	39.00	4.00	9.00	.00	.00	.00	.00
12:												
13:	.00	.00	1.00	.00	1.00	.00	.00	1.00	.00	.00	.00	.00
14:	.00	.00	68.00	11.00	30.00	28.00	.00	2.00	.00	.00	.00	.00
15:												
16:	.00	.00	.00	.00	1.00	.00	.00	.00	.00	.00	.00	.00
17:	.00	.00	107.00	19.00	84.00	68.00	4.00	26.00	.00	.00	.00	.00
18:												
19:	.00	.00	1.00	.00	1.00	.00	.00	1.00	.00	.00	.00	.00
20:	.00	.00	68.00	11.00	30.00	28.00	.00	2.00	.00	.00	.00	.00
21:												
22:	.00	.00	3.00	.00	3.00	2.00	.00	5.00	.00	.00	.00	1.00
23:	.00	.00	108.00	35.00	93.00	95.00	3.00	12.00	.00	.00	.00	.00
24:												
25:	.00	.00	.00	.00	1.00	.00	.00	.00	.00	.00	.00	.00
26:	.00	.00	107.00	19.00	84.00	68.00	4.00	26.00	.00	.00	.00	.00
27:												
28:	.00	.00	1.00	.00	1.00	.00	.00	1.00	.00	.00	.00	.00
29:	.00	.00	68.00	11.00	30.00	28.00	.00	2.00	.00	.00	.00	.00
30:												
31:	.00	.00	.00	.00	1.00	.00	.00	.00	.00	.00	.00	.00
32:	.00	.00	107.00	19.00	84.00	68.00	4.00	26.00	.00	.00	.00	.00
33:												
34:	.00	.00	.00	.00	5.00	2.00	.00	4.00	.00	.00	1.00	.00
35:	.00	.00	128.00	1.00	40.00	39.00	4.00	9.00	.00	.00	.00	.00
36:												
37:	.00	.00	.00	.00	20.00	29.00	.00	25.00	.00	5.00	2.00	1.00
38:	.00	2.00	193.00	2.00	45.00	101.00	32.00	16.00	3.00	.00	.00	.00
39:												
40:	.00	.00	.00	.00	1.00	.00	.00	.00	.00	.00	.00	.00
41:	.00	.00	107.00	19.00	84.00	68.00	4.00	26.00	.00	.00	.00	.00
42:												
43:	.00	.00	1.00	.00	1.00	.00	.00	1.00	.00	.00	.00	.00
44:	.00	.00	68.00	11.00	30.00	28.00	.00	2.00	.00	.00	.00	.00
45:												
46:	.00	.00	.00	.00	1.00	.00	.00	.00	.00	.00	.00	.00
47:	.00	.00	107.00	19.00	84.00	68.00	4.00	26.00	.00	.00	.00	.00
48:												
49:	.00	.00	.00	.00	5.00	2.00	.00	4.00	.00	.00	1.00	.00
50:	.00	.00	128.00	1.00	40.00	39.00	4.00	9.00	.00	.00	.00	.00
51:												
52:	.00	.00	1.00	.00	1.00	.00	.00	1.00	.00	.00	.00	.00
53:	.00	.00	68.00	11.00	30.00	28.00	.00	2.00	.00	.00	.00	.00
54:												
55:	.00	.00	.00	.00	1.00	.00	.00	.00	.00	.00	.00	.00
56:	.00	.00	107.00	19.00	84.00	68.00	4.00	26.00	.00	.00	.00	.00
57:												

Figure III.3.4



UNCLASSIFIED(\*\*\*FILE NAME:WIMP ELEMENT NAME:WIMP/TOE-IN\*\*\*UNCLASSIFIED

```

1:C*****
2:C*****
3:C*****THIS UTILITY PROGRAM IS WRITTEN IN FORTRAN V.
4:C*****
5:C*****
6:C*****PURPOSE OF PROGRAM:
7:C*****
8:C*****THIS UTILITY IS DESIGNED TO CREATE A SEPERATE TOE BY
9:C*****ARTILLERY VULNERABILITY CATEGORIES FOR EACH BLUE UNIT
10:C*****IN A STYLIZED POSTURE'S ARRAY. INPUT DATA FOR THIS
11:C*****UTILITY IS OBTAINED FROM THE "RAM/MATRIX" DATA FILE
12:C*****AND THE "UNITXXX" INPUT DATA FILE TO THE WIMP FOR
13:C*****THE APPROPRIATE POSTURE. THE OUTPUT DATA CREATED BY
14:C*****THIS UTILITY WILL BE USED BY THE WIMP.
15:C*****
16:C*****THIS UTILITY IS ONLY USED INCONJUNCTION WITH THE
17:C*****WIMP.
18:C*****
19:C*****
20:C*****VARIABLE DICTIONARY
21:C*****VARIABLE NAME      DEFINITION
22:C*****
23:C*****NCODE1              USED TO READ THE UNIT CODE
24:C*****                    FROM THE RAM/MATRIX DATA FILE.
25:C*****
26:C*****NCODE2              USED TO READ THE UNIT CODE
27:C*****                    FROM THE WIMP UNIT FILE.
28:C*****
29:C*****NCTY(I,J)           USE: TO READ THE QUANTITY OF
30:C*****                    EQUIPMENT IN EACH VULNER-
31:C*****                    ABILITY CATEGORY.
32:C*****
33:C*****PERS                DUMMY VARIABLE USED TO PLACE
34:C*****                    A ZERO IN VULNERABILITY CATE-
35:C*****                    GORY ONE.
36:C*****
37:C*****
38:C*****
39:      DIMENSION NCODE1(500),NCTY(500,22),QTY(500,22)
40:C*****
41:C*****
42:C*****READING IN THE UNIT CODES AND EQUIPMENT QUANTITIES
43:C*****FROM THE RAM/MATRIX DATA FILE.
44:C*****
45:C*****
46:10      II=II+1
47:      READ(7,700,END=96)NCODE1(II),(NCTY(II,JJ),JJ=1,22)
48:700      FORMAT(3X,12,22I4)
49:      GO TO 30
50:C*****
51:C*****
52:C*****READING IN THE UNIT CODE FROM THE WIMP UNIT DATA
53:C*****FILE.
54:C*****
55:C*****
56:92      KK=0
57:      JL=II-1

```

Figure III.3.6

UNCLASSIFIED\*\*\*FILE NAME:82WIMP ELEMENT NAME:WIMP/TOE-IN\*\*\*UNCLASSIFIED

```

58:      READ(8,800,END=99)INTYPE,NCODE2
59:800    FORMAT(1X,13,3X,12)
60:      IF(INTYPE.GE.500)GO TO 99
61:      IF(INCODE2.EQ.C)GO TO 98
62:C*****
63:C*****
64:C*****CHECKING THE UNIT CODES BETWEEN THE WIMP UNIT FILE
65:C*****AND THE RAM/MATRIX FILE FOR A MATCH AND WRITING
66:C*****OUT THE CORRECT UNIT TOE BY ARTILLERY VULNERABILITY
67:C*****CATEGORY. PERSONNEL IS NOT HANDLED BY THIS UTILITY
68:C*****THUS CATEGORY ONE IS SET TO ZERO.
69:C*****
70:C*****
71:      DO 20 KK=1,JL
72:      IF(INCODE2.NE.NCODE1(KK))GO TO 20
73:      DO 25 MM=1,22
74:      QTY(KK,MM)=NLTV(KK,MM)
75:25    CONTINUE
76:      PERS=0.00
77:      WRITE(9,900)PERS,(QTY(KK,LN),LN=1,11)
78:900    FORMAT(1X,F7.2,11(1X,F7.2))
79:      WRITE(9,901)(QTY(KK,LN),LN=12,22)
80:901    FORMAT(11(1X,F7.2),/)
81:      GO TO 98
82:20    CONTINUE
83:99    STOP
84:      END

```

Figure III.3.6 (Cont)

## Chapter 4

### UTILITY WIMP/LOSS-RATES

**4.1 DESCRIPTION OF PROCESSING:** The WIMP/LOSS-RATES utility program was developed to produce the final product of the WARRAMP Methodology for materiel and the WARF concept. The program's application is contingent upon the utilization of the WIMP in the WARRAMP methodology, as opposed to the AMMO-RATES methodology. The program is implemented in the Fortran V programming language and consists of approximately 40 lines of executable computer source code.

**4.1.1. PURPOSE/FUNCTIONS:** The purpose of the WIMP/LOSS-RATES program is to summarize the losses of blue equipment given N samples (executions of the WIMP) produced from the red artillery engagements on the blue force (one selected posture), and compute the total average losses (quantities) of equipment by artillery vulnerability group, and compute a rate of loss or percentage loss by artillery vulnerability group. These tasks are achieved for the analyst by the program through the accomplishment of the following functions. First the input data (output from the WIMP) is read in and stored in an array sequentially, performing the totaling computation for each artillery vulnerability category. The current methodology specifies 22 groups, plus 1 for personnel; a total of 23. An average loss for the N runs for each vulnerability group is computed and stored in an array. Next the WIMP-XX/TOTAL-CAT data is read by the program. This data is the total number of items per vulnerability category modeled in the force array (for the posture). The losses are computed by simple subtraction of total quantity lost from the total authorized quantity by category. The calculations are performed sequentially and the results written to the respective output files with formatted write statements.

**4.1.2 PROGRAM INPUT/OUTPUT STRUCTURE:** The WIMP/LOSS-RATES I/O structure is depicted in Figure III.4.1. This general graphic presents the input/output filename.elementname normally associated with the program's operation. The four files in each group denote that, at present, the methodology applies four force array postures for a study effort. Also there are multiple (N) copies (samples) of each posture (WIMP-XX-N) input. The number of copies is determined by the analyst statistically (per the number of COSAGE samples) to satisfy statistical confidence requirements.

**4.1.2.A INPUT DATA AND DATA BASE:** The following input data files are required for the program. The normal, demand mode (interactive processing) runstream for the program's execution requires that the data reside in a cataloged mass-storage (disk resident) file. The data and files are required in the order discussed. The runstream, Figure III.4.2, performs two operations on the data prior to program execution. First, the WIMP/XX-N data files are consolidated into one file (logical unit) through the @DATA, I command. Second, the TOTAL-CAT data is placed into a logical unit file via the @DATA, I command.

- o **RUN STREAM data.** Two integer data values are read by the program; both values are contained in the runstream (record 39, Figure III.4.2). The first is the total number of blue units represented in the particular force array, the second is number of the quantity of posture samples, N,

input (number of copies of the WIMP/XX-N file).

- o WIMP/XX-N data. This input data is produced (on output) by the WIMP program discussed in Chapter 2. The file is formatted with decimal (real) data as depicted in figure III.4.3. It is incumbent on the runstream user to edit the runstream and replace the "XX-N" character string with the appropriate posture abbreviation and iteration number, e.g. "AT-9," denoting the attack posture force array data and the 9th run sample.
- o WIMP-XX/TOTAL-CAT data. This input data is produced (on output) by WIMP/TOTAL-CAT program discussed in Chapter 13. The file consists of one (1) record of formatted integer data per artillery vulnerability group plus personnel (23 each) as depicted in figure III.4.4. One file is utilized for each posture modeled in the high resolution ground combat model; however one copy of the appropriate posture is required for program execution.

**4.1.2.B OUTPUT DATA AND DATA FILES:** This program produces two (2) output files, one of which contains the loss data and the second contains the loss rate data. The output is placed into a cataloged file as a file element according to runstream commands. The files are produced by the program in the order discussed.

- o WIMP/LOSS-XX output data and file. The sample output data file is depicted in Figure III.4.5; one record is written in decimal format for each artillery vulnerability category plus personnel. The records and data are positional with personnel losses on record 1 followed by each category on subsequent records. The record format is as follows:

<u>Column</u>	<u>Description</u>	<u>Field</u>
1	Blank	1X
2-11	Quantity of equipment items lost	F10.2

The "XX"Character string is replaced during text editing of the runstream by the user to denote the posture source data to provide a data audit trail.

- o WIMP/RATES-XX output data and file. The sample output is depicted in Figure III.4.6; one record is written in decimal format for each artillery vulnerability category plus personnel. The records and data are positional with personnel loss rates on record 1 followed by the loss rates for each of the modeled categories on subsequent records. The record format is as follows:

<u>Column</u>	<u>Description</u>	<u>Field</u>
1-7	The simple percent of the total equipment lost	F7.5

The "XX" character string is replaced during text editing of the runstream by the user to denote the posture source data to provide a data audit trail.

4.1.2.C DATA ELEMENT DICTIONARY: The data elements used in the program are as follows:

<u>Variable Name</u>	<u>Mode</u>	<u>Value</u>	<u>Definition</u>
AQLOSS(I)	Real	0-23	A 1-dimensional array containing the average loss of equipment of the indexed vulnerability category.
AQTY(I)	Real	0-23	A 1-dimensional array containing the average equipment quantity of the indexed vulnerability category.
ALOSS	Real	--	A variable set to the computed percent of equipment in a category lost.
NAUTH(I)	Integer	0-23	A 1-dimensional array containing the total quantities of the indexed category in the blue units.
NBLUE	Integer	--	A variable set to the input number of blue units modeled
NRUN	Integer	--	A variable set to the input number of run samples considered in the calculations

QTY(I)	Integer	--	A 1-dimensional array containing the quantity of items on-hand by a blue force unit, indexed to the vulnerability categories.
TQTY(I)	Integer	--	A 1-dimensional array containing the total quantity of equipment items on hand, by the blue force units, indexed to the categories.

4.1.3 **PROGRAM PROCESSING:** The program processes are straight forward. The input data is read in, totals are summed, the net loss is computed through subtraction, and the loss rate is computed through division; the computed values are then written onto records in the output file. The program consists of one "routine" element and thus common blocks and other linkages are not used.

4.1.3.A **PROGRAM RUN DESCRIPTION:** The program runstream is depicted in Figure III.4.2. The programs short length enables users to run the program in the demand mode from the terminal. See Volume I, Chapter 4, Section III of this document set for additional details.

4.1.3.B **PROGRAM LOGIC:** The programs logic is graphically presented in Figure III.4.7.

4.1.3.C **PROCESSING FEATURES:** The processing features are straight forward logical tests of the data and can be determined from a review of the program listing Figure III.4.8 and the logic diagram, Figure III.4.7.

4.2 **OPERATING ENVIRONMENT:** The WIMP/LOSS RATES program is compiled on and operated on the USACAA UNIVAC 1100/82 operating system which is a secure system in compliance with Department of the Army security directives. User identification and password codes are required for system access.

4.2.1 **HARDWARE:** The only explicit hardware requirements is that of the UNIVAC 1100/82 OS and a compatible terminal (UTS 400). The operating system dynamically handles on-line storage requirements for the input/output and temporary files. Object code and permanent files for the I/O functions and runstream, must be explicitly cataloged (assigned) to disk-pack files, either removable or fixed. The files may reside on a reserved disk pack or system (public) removable disk pack.

4.2.2 **SUPPORT SOFTWARE:** This program requires the following for demand-mode processing. The program has no Executive-8 release dependent characteristics.

4.2.2.A **SYSTEM SOFTWARE:** The program utilizes the following system processors.

- o ED - text editor processor.
- o MAP - relocatable code collector.
- o DATA - data processor which develops the system data format (SDF) files.

4.2.2.B COMPILER: The operating system library (SYSS\$LIB) contains the FORTRAN V compiler for source program compilation.

4.2.2.C DATA BASE: The data base and files necessary for program execution were discussed in paragraph 4.1.2.A. These files are all Field data in SDF format. These files exist as user cataloged file elements and are not part of a formal data base structure.

4.3 MAINTENANCE PROCEDURES: The relative size of this program minimizes the maintenance on the program. The program is well documented with comment lines (records). Modifications to the program must be clearly denoted by programmers in the manner shown in the listing, Figure III.4.8. Page by page changes to this chapter must be made when changes to the program are made which alter the logic, input or output. the following information is pertinent to the program maintenance (space provided for notes):

Source code Filename.Elementname:

Absolute code Filename.Elementname:

Space required, Source code:

Space required, Object (absolute) code:

Archived tape location:

Read/Write Protect keys: Established by the current custodian.

4.3.1 PROGRAMMING CONVENTIONS: Each function of the program is connected in the listing; standard Fortran techniques are employed with FORTRAN V conventions.

4.3.2 VERIFICATION PROCEDURES: The program is verified by performing test executions (runs) of the program followed by hand calculations and comparisons of the results.

4.3.3 ERROR CORRECTION PROCEDURES: Debugging and error correction is accomplished by an examination of the compiler listing and the run execution (PRINT\$) file.

4.3.4 SPECIAL MAINTENANCE PROCEDURES: None.

# WIMP/LOSS-RATES STRUCTURE

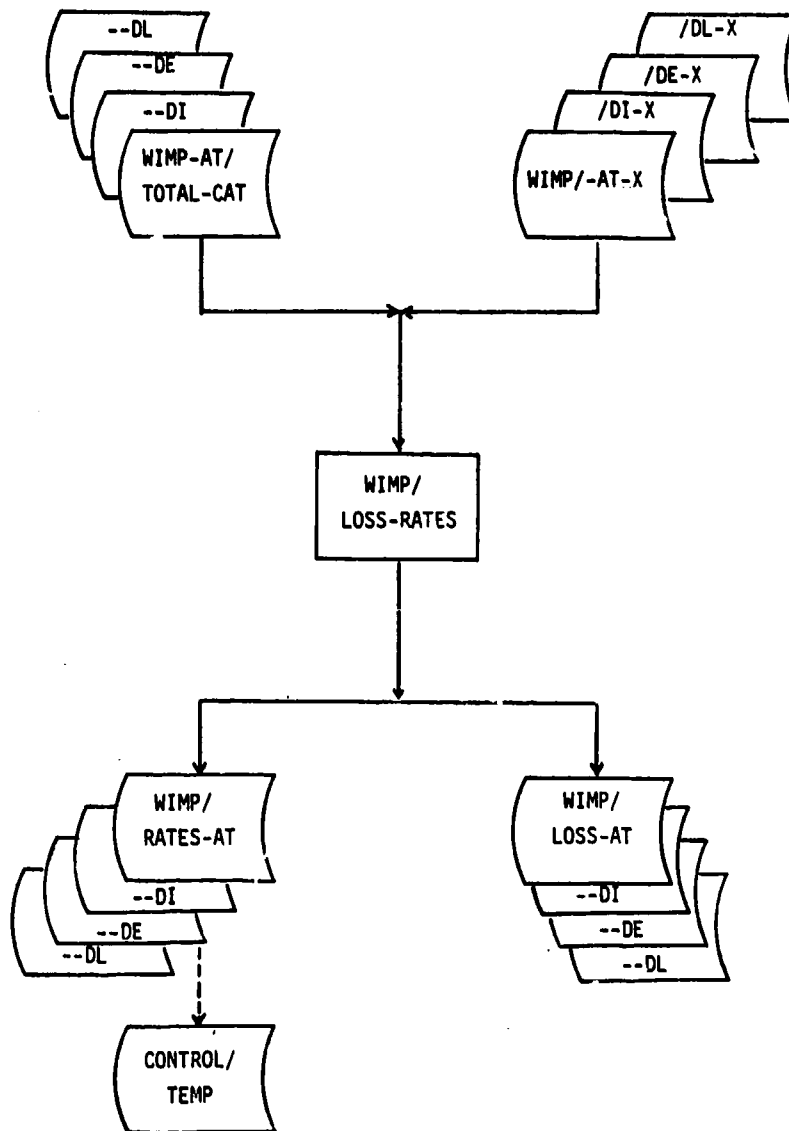


Figure III.4.1

```

1:ERRKPT PRINTS/82PRINT
2:0USF 86.,SECRET*62*ANFP88.
3:0ASG,A 88/ / .
4:0ASG,T 7.
5:0DATA,I 7.
6:0ADD,D 88.WIMP/XX-1
7:0ADD,D 88.WIMP/XX-2
8:0ADD,D 88.WIMP/XX-3
9:0ADD,D 88.WIMP/XX-4
10:0ADD,D 88.WIMP/XX-5
11:0ADD,D 88.WIMP/XX-6
12:0ADD,D 88.WIMP/XX-7
13:0ADD,D 88.WIMP/XX-8
14:0ADD,D 88.WIMP/XX-9
15:0ADD,D 88.WIMP/XX-10
16:END
17:0MSG,N THE ABOVE ELEMENTS ARE THE SEPARATE OUTPUTS FROM
18:0MSG,N EACH RUN OF THE WIMP FOR A SINGLE POSTURE (IE. ATTACK, DEF-
19:0MSG,N ENSE INTENSE, DELAY UP DEFENSE LIGHT). THE "XX"
20:0MSG,N PORTION OF THE ELEMENT NAME MUST BE CHANGED TO REFLECT
21:0MSG,N THE CORRECT POSTURE (IE. AT, DI, DE OR DL). THE NUMBER OF
22:0MSG,N ELEMENT FILES MAY ALSO CHANGE DEPENDING ON THE
23:0MSG,N NUMBER OF TIMES THE WIMP IS EXECUTED FOR ANY ONE
24:0MSG,N POSTURE.
25:0ASG,T 9.
26:0DATA,I 9.
27:0ADD,D 88.WIMP-XX/TOTAL-CAT
28:END
29:0MSG,N THE ABOVE ELEMENT "WIMP/TOTAL-CAT-XX" CONTAINS THE 88
30:0MSG,N DATA FROM UTILITY 82WIMP.WIMP/TOTAL-CAT. THIS UTILITY IS
31:0MSG,N EXECUTED ONCE FOR EACH POSTURE. THE "XX" PORTION OF THE
32:0MSG,N ELEMENT NAME MUST BE CHANGED TO REFLECT THE APPROPRIATE
33:0MSG,N POSTURE (IE. AT, DI, DE OR DL).
34:0ASG,T 8.
35:0ASG,T 10.
36:0ASG,T 11.
37:0ASG,A 82WIMP.
38:0X07 82WIMP.WIMP/LOSS-RATES
39: 189 10
40:0MSG,N THE ABOVE NUMBERS REPRESENT THE NUMBER OF BLUE UNITS IN
41:0MSG,N THE POSTURE'S ARRAY AND THE NUMBER OF TIMES THE WIMP
42:0MSG,N WAS EXECUTED FOR THIS POSTURE. THE FORMAT FOR THIS
43:0MSG,N ENTRY IS COLUMN 1 = BLANK, COLUMNS 2-4 = BLUE UNIT COUNT,
44:0MSG,N COLUMN 5 = BLANK, COLUMNS 6-7 = NUMBER OF TIMES THE WIMP
45:0MSG,N WAS EXECUTED FOR THE POSTURE.
46:0ED 8.,88.WIMP/RATES-XX
47:LNP
48:EXIT
49:0MSG,N THE ABOVE ELEMENT "WIMP/RATES-XX" WILL CONTAIN THE
50:0MSG,N OUTPUT DATA FROM THIS UTILITY. THE "XX" PORTION OF
51:0MSG,N ELEMENT NAME MUST BE CHANGED TO REFLECT THE APPROPRIATE
52:0MSG,N POSTURE (IE. AT, DI, DE OR DL).
53:0ED,R 88.WIMP-XX/TOTAL-CAT
54:LNP
55:EXIT
56:0MSG,N THE ABOVE ELEMENT "WIMP-XX/TOTAL-CAT" CONTAINS THE
57:0MSG,N TOTAL MIE IN EACH EQUIPMENT VULNERABILITY CATEGORY FOR THE
58:0MSG,N ENTIRE POSTURE'S ARRAY. THE "XX" PORTION OF THE
59:0MSG,N ELEMENT NAME MUST BE CHANGED TO REFLECT THE APPROPRIATE
60:0MSG,N POSTURE (IE. AT, DI, DE OR DL).
61:0ED 11.,88.WIMP/LOSS-XX
62:LNP
63:EXIT
64:0MSG,N THE ABOVE ELEMENT "WIMP/LOSS-XX" CONTAINS THE
65:0MSG,N TOTAL MIE LOST IN EACH EQUIPMENT VULNERABILITY CATEGORY
66:0MSG,N FOR THE ENTIRE POSTURE'S ARRAY. THE "XX" PORTION OF THE
67:0MSG,N ELEMENT NAME MUST BE CHANGED TO REFLECT THE APPROPRIATE
68:0MSG,N POSTURE (IE. AT, DI, DE OR DL).
69:0FREE 7.
70:0FREE 8.
71:0FREE 9.
72:0FREE 10.
73:0FREE 11.
74:0FREE 88.
75:0BMMPT PRINTS
76:0ED,R 82PRINT.

```

Figure III.4.2

UNCLASSIFIED\*\*\*EXAMPLE OF OUTPUT DATA OF UTILITY JMP/TOE-1\*\*\*UNCLASSIFIED

1:	.00	.00	.00	.00	1.00	.00	.00	.00	.00	.00	.00	.00
2:	.00	.00	137.00	19.00	94.00	68.00	4.00	26.00	.00	.00	.00	.00
3:	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
4:	.00	.00	1.00	.00	1.00	.00	.00	1.00	.00	.00	.00	.00
5:	.00	.00	68.00	11.00	30.00	28.00	.00	2.00	.00	.00	.00	.00
6:	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
7:	.00	.00	.00	.00	1.00	.00	.00	.00	.00	.00	.00	.00
8:	.00	.00	137.00	19.00	94.00	68.00	4.00	26.00	.00	.00	.00	.00
9:	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
10:	.00	.00	.00	.00	5.00	2.00	.00	4.00	.00	.00	1.00	.00
11:	.00	.00	128.00	1.00	90.00	39.00	4.00	9.00	.00	.00	.00	.00
12:	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
13:	.00	.00	1.00	.00	1.00	.00	.00	1.00	.00	.00	.00	.00
14:	.00	.00	68.00	11.00	30.00	28.00	.00	2.00	.00	.00	.00	.00
15:	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
16:	.00	.00	.00	.00	1.00	.00	.00	.00	.00	.00	.00	.00
17:	.00	.00	137.00	19.00	94.00	68.00	4.00	26.00	.00	.00	.00	.00
18:	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
19:	.00	.00	1.00	.00	1.00	.00	.00	1.00	.00	.00	.00	.00
20:	.00	.00	68.00	11.00	30.00	28.00	.00	2.00	.00	.00	.00	.00
21:	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
22:	.00	.00	3.00	.00	3.00	2.00	.00	5.00	.00	.00	.00	1.00
23:	.00	.00	108.00	35.00	90.00	95.00	3.00	17.00	.00	.00	.00	.00
24:	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
25:	.00	.00	.00	.00	1.00	.00	.00	.00	.00	.00	.00	.00
26:	.00	.00	137.00	19.00	94.00	68.00	4.00	26.00	.00	.00	.00	.00
27:	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
28:	.00	.00	1.00	.00	1.00	.00	.00	1.00	.00	.00	.00	.00
29:	.00	.00	68.00	11.00	30.00	28.00	.00	2.00	.00	.00	.00	.00
30:	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
31:	.00	.00	.00	.00	1.00	.00	.00	.00	.00	.00	.00	.00
32:	.00	.00	137.00	19.00	94.00	68.00	4.00	26.00	.00	.00	.00	.00
33:	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
34:	.00	.00	.00	.00	5.00	2.00	.00	4.00	.00	.00	1.00	.00
35:	.00	.00	128.00	1.00	90.00	39.00	4.00	9.00	.00	.00	.00	.00
36:	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
37:	.00	.00	.00	.00	20.00	29.00	.00	25.00	.00	5.00	2.00	1.00
38:	.00	2.00	193.00	2.00	95.00	101.00	32.00	16.00	3.00	.00	.00	.00
39:	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
40:	.00	.00	.00	.00	1.00	.00	.00	.00	.00	.00	.00	.00
41:	.00	.00	137.00	19.00	94.00	68.00	4.00	26.00	.00	.00	.00	.00
42:	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
43:	.00	.00	1.00	.00	1.00	.00	.00	1.00	.00	.00	.00	.00
44:	.00	.00	68.00	11.00	30.00	28.00	.00	2.00	.00	.00	.00	.00
45:	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
46:	.00	.00	.00	.00	1.00	.00	.00	.00	.00	.00	.00	.00
47:	.00	.00	137.00	19.00	94.00	68.00	4.00	26.00	.00	.00	.00	.00
48:	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
49:	.00	.00	.00	.00	5.00	2.00	.00	4.00	.00	.00	1.00	.00
50:	.00	.00	128.00	1.00	90.00	39.00	4.00	9.00	.00	.00	.00	.00
51:	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
52:	.00	.00	1.00	.00	1.00	.00	.00	1.00	.00	.00	.00	.00
53:	.00	.00	68.00	11.00	30.00	28.00	.00	2.00	.00	.00	.00	.00
54:	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
55:	.00	.00	.00	.00	1.00	.00	.00	.00	.00	.00	.00	.00
56:	.00	.00	137.00	19.00	94.00	68.00	4.00	26.00	.00	.00	.00	.00
57:	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00

Figure III.4.3

UNCLASSIFIED\*\*\*EXAMPLE OF THE OUTPUT DATA OF UTILITY WIMP/TOTAL-CAT\*\*\*UNCLASSIFI

1:	0
2:	336
3:	251
4:	0
5:	1051
6:	718
7:	0
8:	952
9:	6
10:	170
11:	30
12:	37
13:	175
14:	120
15:	25316
16:	2914
17:	10642
18:	10101
19:	1736
20:	3835
21:	91
22:	57
23:	0

Figure III.4.4

UNCLASSIFIED\*\*\*EXAMPLE OF WIMP/LOSS-RATES OUTPUT FILE WIMP/LOSS-XX

1:	.00
2:	.00
3:	4.31
4:	.00
5:	19.44
6:	17.56
7:	.00
8:	1.79
9:	.14
10:	.60
11:	.00
12:	.04
13:	100.15
14:	.00
15:	61.43
16:	20.59
17:	44.11
18:	63.41
19:	19.88
20:	2.20
21:	1.35
22:	.00
23:	.00

Figure III.4.5

UNCLASSIFIED\*\*\*EXAMPLE OF WIMP/LOSS-RATES OUTPUT FILE WIMP/RATES-XY

1: .00000  
2: .00000  
3: .01718  
4: .00000  
5: .01850  
6: .02446  
7: .00000  
8: .00188  
9: .02382  
10: .00499  
11: .00003  
12: .00097  
13: .27141  
14: .00000  
15: .00243  
16: .00707  
17: .00415  
18: .00450  
19: .01133  
20: .00057  
21: .01485  
22: .00000  
23: .00000

Figure III.4.6

QTY (1)  
A QTY (1)  
T QTY (1)  
AQLOSS (1)  
NAUTH (1)

PROGRAM  
WIMP/LOSS-RATFS

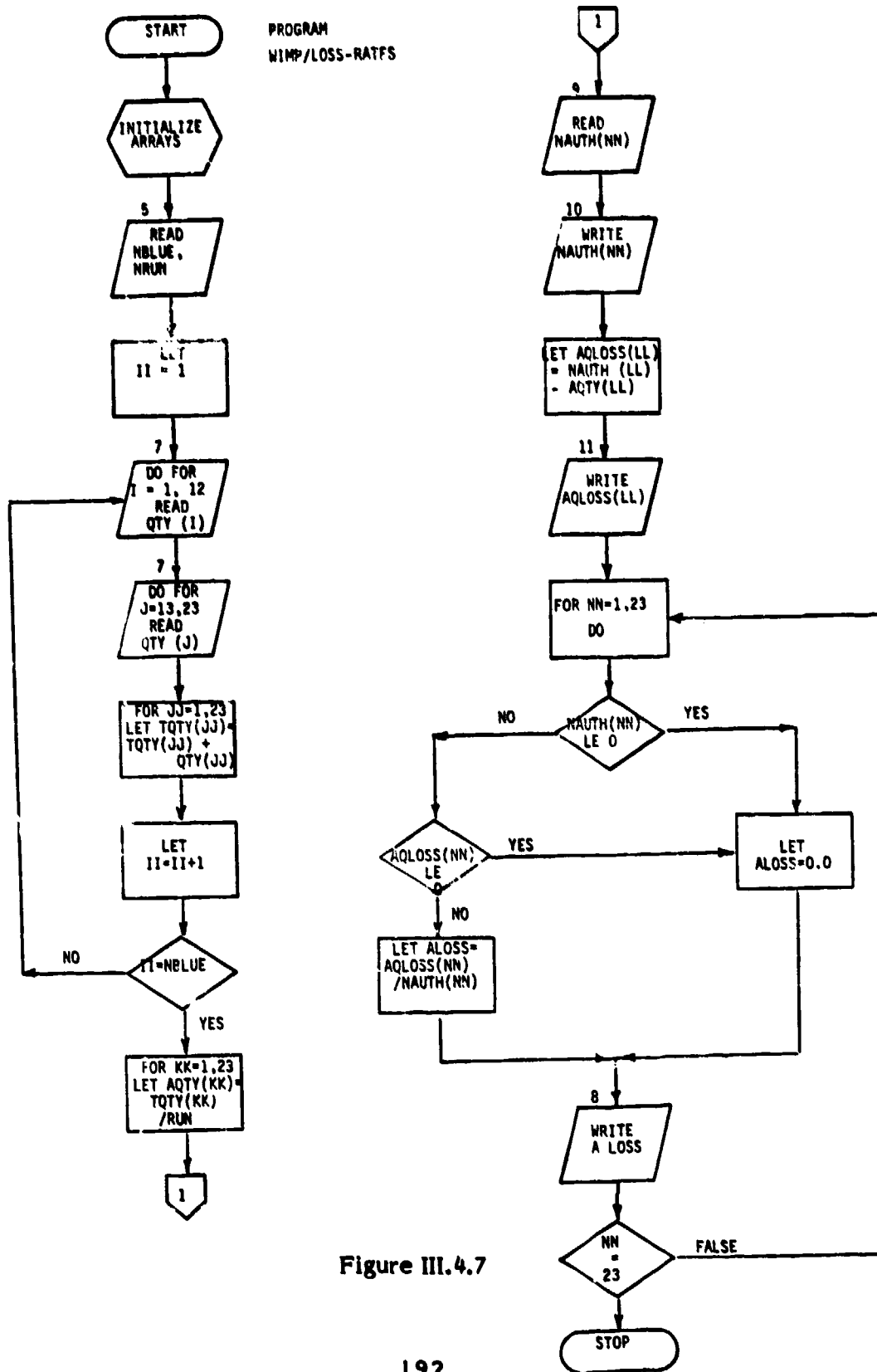


Figure III.4.7

UNCLASSIFIED\*\*\*FILE NAME:WIMP ELEMENT NAME:WIMP/LOSS-RATES\*\*\*UNCLASSIFIED

```

1:C*****
2:C*****
3:C*****THIS UTILITY PROGRAM IS WRITTEN FORTRAN V.
4:C*****
5:C*****
6:C*****PROGRAMMER          DATE          INITIALS
7:C*****
8:C*****CPT SCOTT E CANTLON    JFEN81      SEC
9:C*****
10:C*****
11:C*****
12:C*****PURPOSE OF THE PROGRAM:
13:C*****
14:C*****THIS UTILITY IS DESIGNED TO CONSOLIDATE THE REMAINING
15:C*****QUANTITIES IN EACH BLUE UNIT'S TOE (BY ARTILLERY VUL-
16:C*****NERABILITY CATEGORY) FOR SEVERAL SEPARATE RUNS OF THE
17:C*****WIMP AND PROVIDE A AVERAGE LOSS RATE BY ARTILLERY
18:C*****VULNERABILITY CATEGORY FOR THE ENTIRE ARRAYED POSTURE.
19:C*****
20:C*****
21:C*****
22:C*****VARIABLE DICTIONARY
23:C*****VARIABLE NAME      DEFINITION
24:C*****
25:C*****NBLUE              NUMBER OF BLUE UNITS IN THE ARRAY
26:C*****
27:C*****NRUN              NUMBER OF TIMES THE WIMP WAS EXECUTED
28:C*****                  FOR THE POSTURE.
29:C*****
30:C*****QTY(I)            QUANTITY OF ITEMS REMAINING IN A UNIT'S
31:C*****                  ARTY VULNERABILITY CATEGORIES AFTER
32:C*****                  EXECUTION OF THE WIMP.
33:C*****
34:C*****TOTY(I)           TOTAL QUANTITY OF ITEMS REMAINING IN THE
35:C*****                  ARRAY BY ARTY VULNERABILITY CATEGORY.
36:C*****
37:C*****ACTY(I)           AVERAGE QUANTITY OF ITEMS REMAINING IN THE
38:C*****                  ARRAY BY ARTY VULNERABILITY CATEGORY FOR
39:C*****                  SEVERAL RUNS OF THE WIMP.
40:C*****
41:C*****AQLOSS(I)         AVERAGE QUANTITY OF ITEMS LOSS IN EACH
42:C*****                  CATEGORY FOR ALL RUNS OF THE WIMP FOR
43:C*****                  A POSTURE.
44:C*****
45:C*****NAUTH(I)          TOTAL AUTHORIZED QUANTITY FOR EACH CATEGORY
46:C*****                  FOR THE ENTIRE ARRAY.
47:C*****
48:C*****ALOSS             AVERAGE PERCENT OF LOSS FOR EACH CATEGORY
49:C*****                  FOR THE ENTIRE ARRAY.
50:C*****
51:C*****TITLE(1-3)        VULNERABILITY CATEGORY TITLES.
52:C*****
53:C*****
54:C*****
55:      DIMENSION QTY(23),ACTY(23),TOTY(23),AQLOSS(23),NAUTH(23),
56:      *TITLE1(23),TITLE2(23),TITLE3(23)
57:C*****
58:C*****READING IN THE VULN CAT TITLES.
59:C*****
60:      7C
61:      READ (12,127,END=15)TITLE1(J),TITLE2(J),TITLE3(J)
62:1200  FORMAT(A6,A1,A6)
63:      GO TO 7C
64:C*****
65:C*****READING IN THE NUMBER OF BLUE UNITS IN THE ARRAY
66:C*****AND THE NUMBER OF TIMES THE WIMP WAS EXECUTED.

```

Figure III.4.8

```

67:C*****
68:15 READ(15,500,FND=10)NBLUE,NRUN
69:500 FORMAT(1X,13,1X,12)
70:C*****
71:C*****READING INTO AN ARRAY THE QUANTITY OF ITEMS REMAINING
72:C*****EACH CATEGORY FOR EACH UNIT.
73:C*****
74:10 DO 20 II=1,NBLUE
75: READ(7,700,FND=98)(QTY(II),I=1,12)
76:700 FORMAT(12F7.2)
77: READ(7,701,FND=98)(QTY(IJ),J=13,23)
78:701 FORMAT(11F7.2,/)
79:C*****
80:C*****TOTALING UP ALL THE REMAINING ITEMS FOR EACH CATEGORY
81:C*****AND ALL RUNS OF THE WIMP.
82:C*****
83: DO 25 JJ=1,23
84: TOTY(JJ)=TOTY(JJ)+QTY(IJ)
85:25 CONTINUE
86:20 CONTINUE
87: GO TO 15
88:C*****
89:C*****COMPUTING THE AVERAGE QUANTITY REMAINING IN EACH CATEGORY
90:C*****FOR THE ENTIRE ARRAY.
91:C*****
92:98 DO 30 KK=1,23
93: ACTY(KK)=TOTY(KK)/NRUN
94:30 CONTINUE
95:C*****
96:C*****READING IN THE TOTAL AUTHORIZED QUANTITY OF ITEMS IN
97:C*****EACH CATEGORY FOR THE ENTIRE ARRAY.
98:C*****
99: DO 40 MM=1,23
100: READ(9,900,FND=99)NAUTH(MM)
101:900 FORMAT(1I7)
102: WRITE(10,1000)NAUTH(MM),TITLE1(MM),TITLE2(MM),TITLE3(MM)
103:1000 FORMAT(1X,11,5X,A6,A6,A6)
104:40 CONTINUE
105:C*****
106:C*****COMPUTING THE AVERAGE NUMBER OF ITEMS LOSS IN EACH
107:C*****CATEGORY FOR THE ENTIRE ARRAY.
108:C*****
109:99 DO 50 LL=1,23
110: AQLOSS(LL)=NAUTH(LL)-ACTY(LL)
111: WRITE(11,1100)AQLOSS(LL),TITLE1(LL),TITLE2(LL),TITLE3(LL)
112:1100 FORMAT(1X,F10.2,5X,A6,A6,A6)
113:50 CONTINUE
114:C*****
115:C*****COMPUTING THE LOSS RATE AS A PERCENT FOR EACH CATEGORY
116:C*****AND WRITING OUT THE RESULTS.
117:C*****
118: DO 60 NN=1,23
119: IF(NAUTH(NN).LE.0)GO TO 61
120: IF(AQLOSS(NN).LE.0)GO TO 61
121: ALOSS=AQLOSS(NN)/NAUTH(NN)
122: WRITE(16,800)ALOSS,TITLE1(NN),TITLE2(NN),TITLE3(NN)
123:800 FORMAT(F7.5,5X,A6,A6,A6)
124: GO TO 60
125:61 ALOSS=0.000
126: WRITE(16,800)ALOSS,TITLE1(NN),TITLE2(NN),TITLE3(NN)
127:60 CONTINUE
128: STOP
129: END

```

Figure III.4.8 (Cont)

## CHAPTER 5

### UTILITY: LEA/TAPE

**5.1 DESCRIPTION OF PROCESSING:** This utility program performs no computations; the program is read and write statements.

**5.1.1 PURPOSE/FUNCTION:** The purpose of this utility is to process a magnetic tape supplied by the Logistics Evaluation Agency (LEA) which describes the Major Items of Equipment (MIE) being used in the current study as to the quantity and density profiles of each item by its Line Code (LINC CODE). The utility will split the quantity and density profiles for each Line Code and write them into separate sequential files; one for quantity profile and one for density profile. These files will be used as input to the utility ITMID/TEMP.

**5.1.2 PROGRAM INPUT/OUTPUT STRUCTURE:** The overall structure of the utility is pictured in FIGURE 5.3.1. The general logic flow is pictured in the flow chart, FIGURE III.5.2. The source code can be seen in FIGURE III.5.3.

**5.1.2.A INPUT DATA AND DATA BASE:** The LEA/TAPE, provided by the Logistics Evaluation Agency (LEA), is the only input required by this utility. The format of the tape is depicted in Figure III.5.4.

**5.1.2.B OUTPUT DATA AND DATA FILES:** - The utility produces two mass storage file outputs. One file is the DENSITY/PROFILES file. This file describes each Major Item of Equipment (MIE), identified by its' LINC CODE, its density within each of the five combat zones, for each of the seven time periods being played. Density as described here is defined as the percentage of the available item of equipment which can be found in a combat zone during a specific time period. In all cases, the percentages for an item during a time period must sum to 100%. Thus, by multiplying the quantity of the item available for a given time period as described in the QUANTITY/PROFILES file, by the density percentage for a combat zone for that time period, the actual number of these items in this zone can be determined.

Figure III.5.5 presents a file layout for the DENSITY/PROFILES file and presents an example of the data as it is found in the file. As can be seen from the file layout and the example, there will be five records for each MIE in the file; one record for each combat zone of the battle area. Further, within each record there will be seven occurrences of the actual density percentage of this item found within the combat zone; one for each of the seven time periods of the exercise.

The second mass storage file produced is the QUANTITY/PROFILES file. This file denotes for each MIE as identified by its LINC CODE. The quantity of this item that will be available to be played in each of the seven time periods of the exercise. As can be seen from the record layout in the data example, there will be one record for each MIE and seven occurrences of its quantity; one for each time period. Figure III.5.6 presents an example of the data that can be found in the file.

### 5.1.2.C DATA ELEMENT DICTIONARY:

The following section will identify and define all variables used in this utility.

<u>NAME</u>	<u>DEFINITION</u>
DEN	Density Profiles. This is an array of 7 members; one member for each of the 7 time periods of the study. Further, there will be 5 occurrences of this array for each of the 5 combat zones of the battle area. Each entry in the array will denote the percentage of this line item found in a specific zone during a specific time period. The summation of percentages for a specific zone over the 7 periods must equal 100%.
LIN	Line Code. The six character alphanumeric code assigned to an item of equipment which uniquely identifies the item.
QUAN	Quantity Profile. This is a 7 member array. There will be one occurrence of this array per item. Each entry in the array denotes the quantity of this item authorized for each of the 7 time periods of the study.
I, J, K and L	These 4 variables are used as subscripts in the program.

5.2 OPERATING ENVIRONMENT: This program is implemented with the UNIVAC Executive - 8 operating system.

5.2.1 SUPPORT SOFTWARE: This routine requires the FORTRAN IV compiler and UNIVAC 1100/82 system facilities.

5.2.2 I/O DEVICES: This utility will use one tape drive for its' input file. Its output will be written to disk. Refer to Volume I for the program runstream.

5.3 MAINTENANCE PROCEDURES: The length of this program (16 lines of code) has not warranted explicit maintenance procedures.

5.3.1 PROGRAMMING CONVENTIONS: Standard FORTRAN conventions.

5.3.2 INTERNAL ERROR ROUTINES: None are incorporated in this utility program.

# LEA/TAPE STRUCTURE

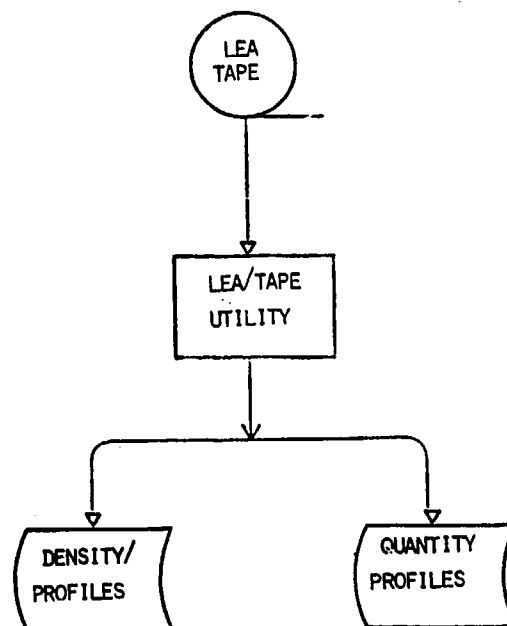


Figure III.5.1

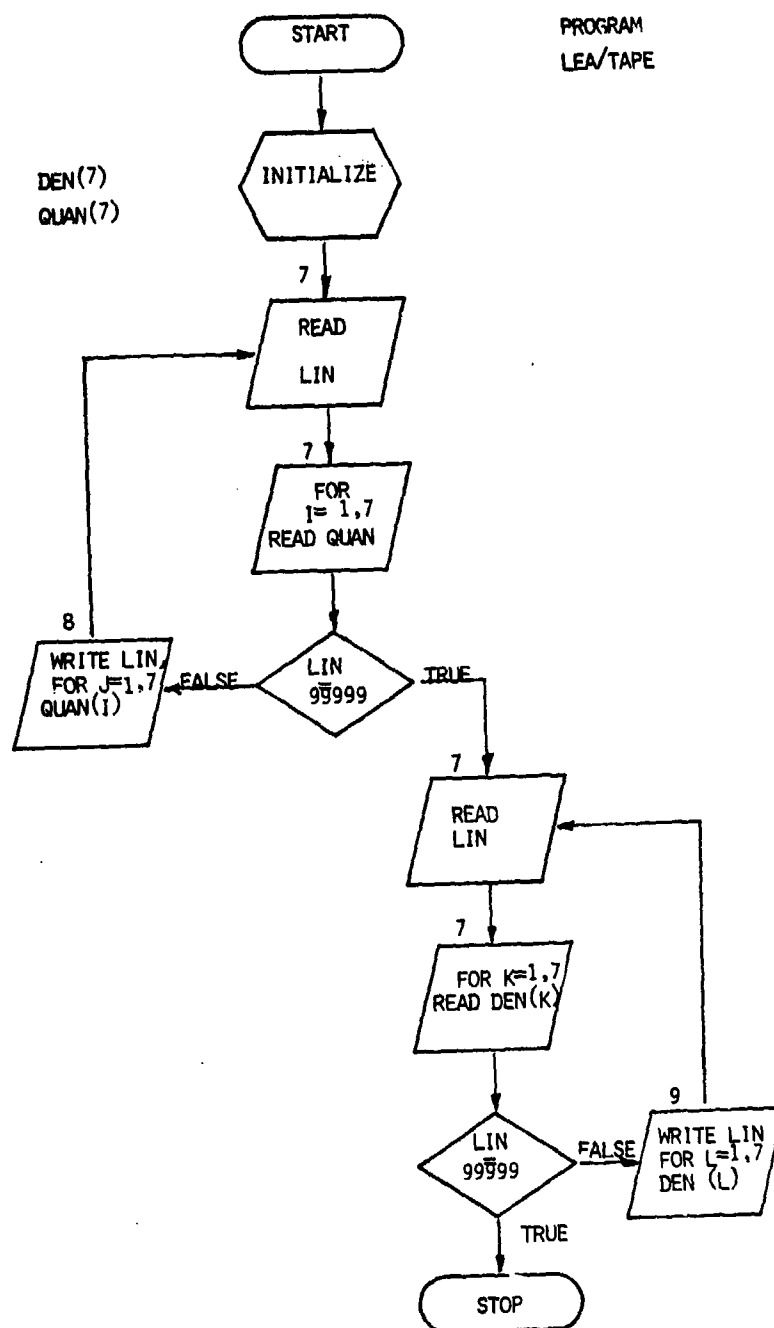


Figure III.5.2

UNCLASSIFIED\*\*\*FILE NAME:82XQT ELEMENT NAME:LEA/TAPE\*\*\*UNCLASSIFIED

```

1:C*****THIS UTILITY IS DESIGNED TO TAKE THE LEA DATA TAPE WITH
2:C*****THE LIN CODE DENSITY AND QUANTITY PROFILES FOR THE CURRENT
3:C*****WAF STUDY AND PLACE THEM IN SEPARATE ELEMENTS OF A
4:C*****PROGRAM FILE USED TO STORE DATA FOR FURTHER MANIPULATE AND
5:C*****RECORD.
6:C*****
7:C*****
8:C*****VARIABLE DICTIONARY
9:C*****VARIABLE NAME      DEFINITION
10:C*****LIN                LIN CODE
11:C*****QUAN(I)             QUANTITY PROFILES
12:C*****DEN(I)              DENSITY PROFILES
13:C*****
14:C*****
15:      DIMENSION DEN(7)
16:      INTEGER QUAN (7)
17:      REAL LIN
18:10      READ(7,700,END=20)LIN,(QUAN(I),I=1,7)
19:700      FORMAT(1X,A6,7(2X,I7))
20:      IF(LIN.EQ.999999)GO TO 20
21:      WRITE (8,800)LIN,(QUAN(J),J=1,7)
22:800      FORMAT(1X,A6,10X,7I9)
23:      GO TO 10
24:20      READ(7,701,END=99)LIN,(DEN(K),K=1,7)
25:701      FORMAT(1X,A6,7(2X,F5.1))
26:      IF(LIN.EQ.999999)GO TO 99
27:      WRITE (9,900)LIN,(DEN(L),L=1,7)
28:900      FORMAT(1X,A6,11X,7F5.1)
29:      GO TO 20
30:99      STOP
31:      END

```

Figure III.5.3

UNCLASSIFIED\*\*\*EXAMPLE OF THE LINCDE/LIST DATA FILE\*\*\*UNCLASSIFIED

```

1: A03198 AK VEH M218 GM EQ P1A
2: A14752 ADAP TEST CAMERA LM178
3: A22496 AIMING CIRCLE M2 W/E
4: A23770 AIR COND FL/WNDW 6000B
5: A23828 AIR COND F/WA 9000 BTU
6: A24044 AIR COND 13000 BTU
7: A24318 AIR COND 18000 BTU
8: A24455 AIR COND FM AIR-COOL
9: A24463 AIR COND F/WA 18000 BT
10: A24592 AIR COND 19000 BTU
11: A24763 AIR COND F/WA 36000BTU
12: A24900 AIR COND 36000 BTU
13: A27159 AIR TRF C F AN/TSQ-97
14: A32444 ALARM CHL AGT M11
15: A32508 ALARM CHL AGT M12
16: A32564 ALARM CHL AGT AUTO
17: A32568 ALARM CHL AGT AUTO
18: A32570 ALARM CHL AGT AUTO
19: A34457 ALGHT FX MX-8409/AAS24
20: A41666 RDR SET AN/TPQ-37 LP
21: A55293 ANAL CHG B AN/ASM-137
22: A55300 ANAL CHG B AN/ASM-490
23: A55304 ANAL CHG B AN/GSM-261
24: A55704 ANAL FLT LN AN/ASM-80
25: A56235 ANAL SET LS-89A
26: A56243 ANAL SET ENG PTBL S S
27: A56800 ANAL SPTCH AN/UPM-58
28: A56937 ANAL SPTCH AN/UPM-84
29: A58033 ANAL SPTCH TS-723/U
30: A77877 ANTENNA GRP AN/GRA-4
31: A79014 ANTENNA GRP AN/GRA-12
32: A79151 ANTENNA GRP AN/GRA-50
33: A99943 ATTENUTR VAR CN-1035/G
34: B01756 AUGER EARTH SM4A
35: B11795 BOTTLE CLEANG AN/TAM-4
36: B18648 BAKERY PLT M-1945LP
37: B30238 BARCE ASSY SET 5X12
38: B30923 BARGE DECK CGO NP OC
39: B43663 BATH U PTBL GED LP
40: B45597 BTRY CHGR PP7286% G/U
41: B51098 BEACON SET AN/TRN-30V1
42: B51099 BEACON SET AN/TRN-30V2
43: B63711 BIN STO AGGR PTBL 60 T
44: B67423 BINOCULAR EL AN/PAS-5
45: B67492 BINOCULAR INFRARED
46: B83582 BOAT BRDG ERECT GD 27
47: B83856 BOAT LAND INFLT 15 MAN
48: B84404 BOAT RECON PNEU 3-MAN
49: C18481 BREAKER PAV-DRILL
50: C20414 BRIDGE ARMD VEH
51: C22058 BRIDGE ERECT SET FIX
52: C22120 BRIDGE ERECT SET FB UK
53: C22911 BRIDGE FXD HI-WAY ALUM
54: C23017 BRIDGE FXD HI-WAY
55: C25757 BRIDGE FLTG RAFT LT
56: C26031 BRIDGE EREC SET
57: C36120 BLUZR EM F/M60 SER TKS

```

Figure III.5.4

UNCLASSIFIED\*\*\*EXAMPLE DENSITY/PROFILE OUTPUT DATA FROM THE LFA/TAPE UTILITY\*\*\*U

1:	A03198	.0	.0	.0	.0	.0	.0
2:	A03198	.0	.0	.0	.0	.0	.0
3:	A03198	.0	.0	.0	.0	.0	.0
4:	A03198	60.5	60.5	60.5	74.6	74.6	74.6
5:	A03198	79.5	39.5	39.5	25.4	25.4	25.4
6:	A14752	.0	.0	.0	.0	.0	.0
7:	A14752	.0	.0	.0	.0	.0	.0
8:	A14752	36.7	36.7	36.7	24.3	24.3	34.3
9:	A14752	63.3	63.3	63.3	75.7	75.7	65.7
10:	A14752	.0	.0	.0	.0	.0	.0
11:	A22496	4.9	6.0	1.6	3.7	3.7	3.7
12:	A22496	13.4	16.6	20.4	20.8	20.8	20.8
13:	A22496	80.0	75.4	66.9	64.5	64.5	64.5
14:	A22496	1.5	11.5	11.1	21.0	21.6	21.6
15:	A22496	.0	.0	.0	.0	.0	.0
16:	A23770	.0	.0	.0	.0	.0	.0
17:	A23770	.0	.0	.0	.0	.0	.0
18:	A23770	.0	.0	.0	.0	.0	.0
19:	A23770	.0	.0	.0	.0	.0	.0
20:	A23770	.0	.0	.0	.0	.0	.0
21:	A23828	.0	.0	.0	.0	.0	.0
22:	A23828	10.0	10.0	10.0	10.0	10.0	10.0
23:	A23828	25.0	23.2	26.2	30.8	30.8	30.8
24:	A23828	10.1	18.0	16.4	17.6	17.6	17.6
25:	A23828	54.9	48.8	45.4	41.5	41.5	41.5
26:	A24044	.0	.0	.0	.0	.0	.0
27:	A24044	.0	.0	.0	.0	.0	.0
28:	A24044	20.0	20.0	20.0	.0	.0	.0
29:	A24044	36.2	36.2	36.2	50.0	50.0	50.0
30:	A24044	43.8	43.8	43.8	50.0	50.0	50.0
31:	A24318	.0	.0	.0	.0	.0	.0
32:	A24318	.0	.0	.0	.0	.0	.0
33:	A24318	.0	.0	.0	.0	.0	.0
34:	A24318	50.0	50.0	60.0	70.0	70.0	80.0
35:	A24318	50.0	50.0	40.0	30.0	20.0	20.0
36:	A24455	.0	.0	.0	.0	.0	.0
37:	A24455	10.0	10.0	10.0	10.0	20.0	30.0
38:	A24455	30.0	30.0	30.0	40.0	40.0	50.0
39:	A24455	60.0	60.0	60.0	50.0	40.0	20.0
40:	A24455	.0	.0	.0	.0	.0	.0
41:	A24463	.0	.0	.0	.0	.0	.0
42:	A24463	.0	.0	.0	.0	.0	.0
43:	A24463	40.5	35.4	39.7	45.7	44.9	44.9
44:	A24463	36.4	42.9	44.8	40.2	40.7	40.7
45:	A24463	23.1	21.7	15.5	14.2	14.3	14.3
46:	A24592	.0	.0	.0	.0	.0	.0
47:	A24592	.0	.0	.0	.0	.0	.0
48:	A24592	50.5	50.8	52.5	55.4	55.4	55.4
49:	A24592	42.7	42.9	41.6	44.3	44.3	44.3
50:	A24592	6.8	6.2	5.9	.3	.3	.3
51:	A24763	.0	.0	.0	.0	.0	.0
52:	A24763	.0	.0	.0	.0	.0	.0
53:	A24763	100.0	100.0	100.0	100.0	100.0	100.0
54:	A24763	.0	.0	.0	.0	.0	.0
55:	A24763	.0	.0	.0	.0	.0	.0
56:	A24300	.0	.0	.0	.0	.0	.0
57:	A24300	.0	.0	.0	.0	.0	.0

Figure III.5.5

UNCLASSIFIED\*\*\*EXAMPLE OF THE QUANTITY/PROFILE DATA FROM UTILITY LEA/TAPP

1: A03198	44	44	44	44	49	48	48	49
2: A14752	4	4	6	6	7	7	7	7
3: A22496	2338	2339	2909	3720	4923	4823	4823	4923
4: A23770	0	0	0	0	0	0	0	0
5: A23828	339	339	403	488	818	818	818	818
6: A24044	10	10	13	13	19	18	15	15
7: A24318	4	4	8	15	57	63	63	67
8: A24455	431	431	788	916	1554	1280	1272	1272
9: A24463	833	833	787	943	1184	1500	1500	1200
10: A24592	44	44	44	43	87	57	57	57
11: A24763	5	5	5	2	2	2	2	2
12: A24900	554	554	653	744	889	993	9024	1035
13: A27159	38	38	39	19	47	49	84	54
14: A32444	3441	3441	3153	2718	3843	3476	3722	3766
15: A32508	540	540	534	833	677	677	677	677
16: A32564	24	32	43	986	535	535	235	235
17: A32568	3845	3845	5267	2966	3779	3810	3834	3936
18: A32570	530	530	440	609	954	924	924	924
19: A34457	4	4	6	6	7	7	7	7
20: A55293	311	311	148	198	543	543	243	243
21: A55300	63	68	73	93	133	133	133	133
22: A55304	4	4	4	6	6	6	6	6
23: A55704	4	4	6	6	7	7	7	7
24: A56235	8	9	9	5	4	6	6	6
25: A56243	3344	3344	4038	8491	7158	7510	7236	7257
26: A56800	58	58	48	84	56	56	56	56
27: A56937	83	83	48	75	86	86	86	86
28: A59033	535	535	279	324	792	404	406	406
29: A77877	84	84	84	84	84	84	84	84
30: A79014	30	30	30	30	30	30	30	30
31: A78151	508	509	523	233	249	280	250	250
32: A90118	837	837	947	9396	1839	1845	1845	1842
33: A90123	348	348	564	564	344	344	344	344
34: A90344	0	0	0	0	0	0	0	0
35: A99943	4	4	4	4	4	4	4	4
36: B01756	48	48	87	33	151	151	121	121
37: B07752	84	84	43	68	68	68	68	68
38: B11795	37	37	955	184	194	194	194	194
39: B18648	33	33	44	46	89	68	74	74
40: B30238	8	8	35	15	12	12	12	12
41: B70923	3	3	5	5	2	2	2	2
42: B43663	377	377	514	522	310	327	330	330
43: B51098	84	84	79	73	915	116	156	126
44: B51099	53	53	49	80	89	88	91	91
45: B63711	54	54	38	47	63	61	61	61
46: B67423	314	314	116	116	150	150	120	120
47: B67492	415	415	415	412	412	412	412	412
48: B83582	8	8	8	8	8	8	8	8
49: B83856	389	389	589	443	1510	1237	1288	1255
50: B94404	480	480	760	3559	1761	1790	1793	1793
51: C18431	3	3	33	39	39	39	39	39
52: C20414	233	233	247	348	923	524	524	524
53: C22058	3	3	7	34	14	14	14	14
54: C22126	12	13	14	14	16	16	16	16
55: C22811	23	24	28	32	32	32	32	32
56: C23017	3	3	7	34	14	14	14	14
57: C25757	60	67	82	93	102	107	107	107

Figure III.5.6

## CHAPTER 6

### UTILITY - ITMID/TEMP

**6.1 DESCRIPTION OF PROCESSING:** This program performs one computation; the balance of the statements are read and write.

**6.1.1 PURPOSE/FUNCTIONS:** The purpose of this utility is to merge together data from the QUANTITY/PROFILES file and the DENSITY/PROFILES created by the previous utility LEA/TAPE with the LINCONE/LIST file. The LINCONE/LIST file is created manually in the LINCONE/LIST element of the WARF program file of the current study, using as its source, data provided by the study's sponsor. The utility reads in the LINCONE and the nomenclature from the LINCONE/LIST file and the corresponding 7-period quantities from the QUANTITY/PROFILE file and the first 7 time period density figures from the DENSITY/PROFILE file and immediately writes out this data onto the ITMID/TEMP file.

The utility does not check to ensure that the LINCONE on the LINCONE/LIST corresponds to the LINCONE on the Quantity Profile and Density Profile files. Therefore, it assumes that all files are sorted on LINCONE and all files will have matching LINCONES.

**6.1.2 PROGRAM INPUT/OUTPUT STRUCTURE:** The overall structure of the utility is pictured in FIGURE III.6.1. The general logic flow within the utility is displayed in the flow chart in FIGURE III.6.2. A copy of the source code is found in FIGURE III.6.3.

**6.1.2.A INPUT DATA AND DATA BASE:** The utility will use these files stored on mass storage devices as input; DENSITY/PROFILE, QUANTITY/PROFILE and the LINCONE/LIST files. Figure III.6.4 gives the file layout and examples of the data for the DENSITY/PROFILES file. Figures III.6.5 does the same for the QUANTITY/PROFILES file and III.6.6 for the LINCONE/LIST file.

**6.1.2.B OUTPUT DATA AND DATA FILES:** This utility produces one output file; the ITMID/TEMP file on mass storage. This file is a multi-record file. The header or identification record identifies the item being described by its LINCONE and provides its nomenclature. The second record provides seven quantity occurrences for this item; one for each of the time periods in the exercise. The last record type will occur seven times; once for each of the combat zones within the battlefield. Within each of these seven records there will be five occurrences of the density distribution for this item once for each of the zones within the battlefield. As in the DENSITY/PROFILE file these densities within zones are percentages of the total equipment available for distribution during a time period. Therefore, the summation over each of the five combat zones for each time period should equal 100% (rounding may produce imprecise values).

Figure III.6.7 gives an example of the ITMID/TEMP output data as it can be found in the file.

### 6.1.2.C DATA ELEMENT DICTIONARY:

The following section will identify and define all variables used in this utility.

<u>NAME</u>	<u>DEFINITION</u>
LIN	Line Code. The 6 character alphanumeric code assigned to an item of equipment which uniquely identifies the item.
NOMEN	A 30 character alphanumeric field which provides a description of the item. The program in constructing this variable uses a one-dimension array of 5 members, each member 6 characters in length. This data will be supplied from the LINCODE/LIST file.
QUAN	Quantity Profile. This is a 7 member array. Each entry in this array for an item will denote the quantity of this item authorized for a specific time period of the study.
DENS	Density Profile. A two-dimension array. The first dimension identifies each of the 5 combat zones in the battle area. The second identifies each of the 7 time periods of the study. Each entry in this array, as it is read in, specific time period. However, before writing this array out to be the ITMID/TEMP file the utility divides the percentage value by 100 in order to express the densities as fractions rather than percentages.
J, K, L, M, & N	Inter values that are used as subscripts in arrays.

6.2 OPERATING ENVIRONMENT: This program is implemented on the UNIVAC Executive-8 operating system.

6.2.1 SUPPORT SOFTWARE: This routine requires the FORTRAN IV compiler and the UNIVAC 1100/82 system facilities.

6.2.2 I/O DEVICES: This utility will use input files resident on disk. It's output will also be written to disk. Refer to Volume I for the execution runstream.

6.3 MAINTENANCE PROCEDURES: There are no explicit or unique procedures applied.

6.3.1 PROGRAMMING CONVENTIONS: Standard FORTRAN conventions.

6.3.2 INTERNAL ERROR ROUTINES: There are no explicit error handling procedures written into this utility.

ITMID/TEMP PROGRAM FLOW

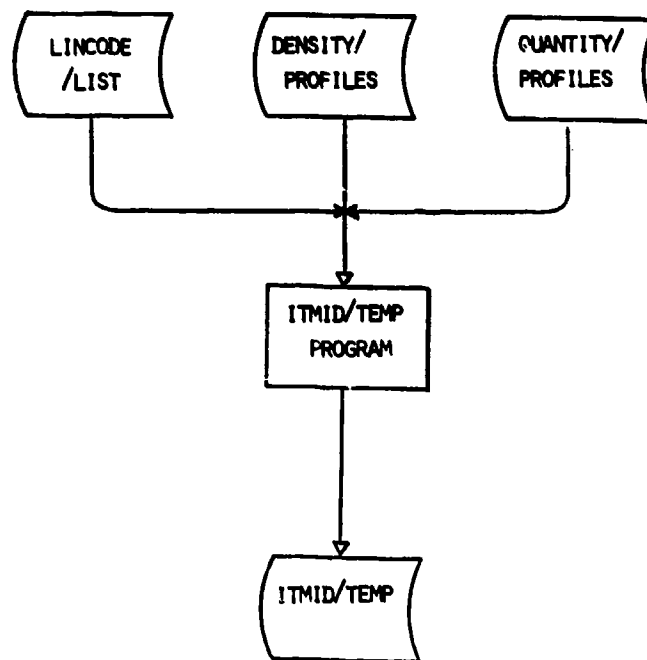


Figure III.6.1

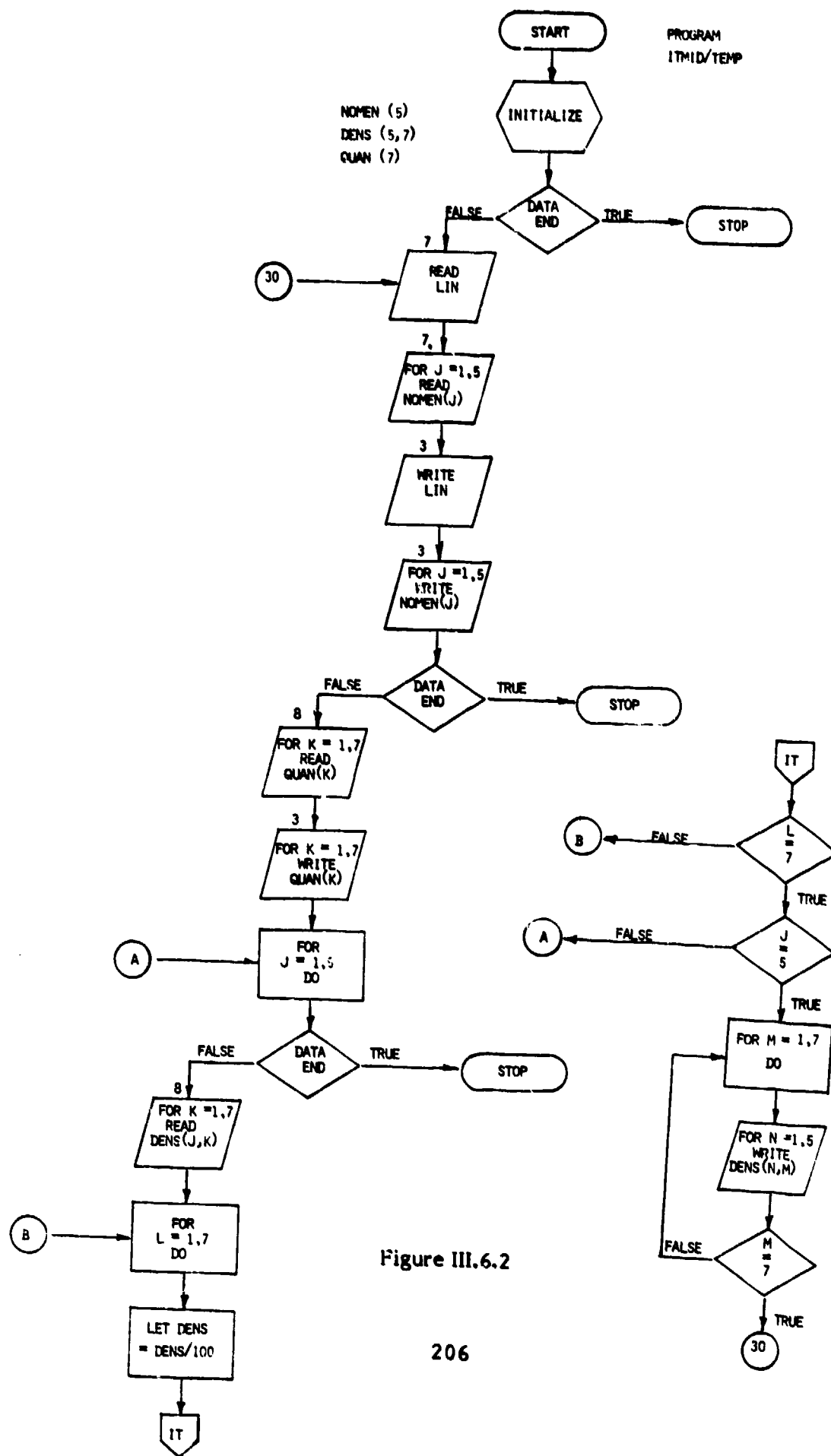


Figure III.6.2

UNCLASSIFIED\*\*\*FILE NAME:82X07 ELEMENT NAME:ITHID/TMP\*\*\*UNCLASSIFIED

```

1:C*****THIS UTILITY PROGRAM IS DESIGNED TO MERGE DATA FROM
2:C*****THE DENSITY PROFILES FILE AND QUANTITY PROFILES FILE CREATED
3:C*****BY THE UTILITY PROGRAM "82X07.LEA/TAPF" AND THE WARF STUDY'S
4:C*****LIN CODE LIST FILE. THE LIN CODE LIST FILE IS CREATED
5:C*****FROM THE PAPER COPY LIST PROVIDED BY THE STUDY SPONSOR. THIS
6:C*****IS NORMALLY DONE BY HAVING THE LIST TYPED INTO PUNCHCARD FORM AND
7:C*****SUBMITTED AS A JOB PLACING IT INTO THE ELEMENT "LINGCODE/LIST" OF
8:C*****THE WARF PROGRAM FILE FOR THE CURRENT STUDY. THIS UTILITY RESULTS IN
9:C*****THE CREATION OF A TEMPORARY ITHID FILE THAT WILL BE FURTHER REFINED
10:C*****BY THE UTILITY "82X07.ITHID/REC-A" AND PRODUCE THE FINAL
11:C*****ITHID FILE TO BE USED BY ELCON.
12:C*****
13:C*****VARIABLE NAME DICTIONARY
14:C*****VARIABLE NAME      DEFINITION
15:C*****LIN                LIN CODE OBTAINED FROM THE WARF DATA FILE.
16:C*****NOMEN              NOMENCLATURE FOR A LIN OBTAINED FROM THE WARF DATA FILE.
17:C*****QUAN              QUANTITY BY LIN FOR EACH TIME PERIOD OBTAINED FROM
18:C*****                  QUANTITY/PROFILE DATA FILE.
19:C*****DENS              DENSITY FOR A LIN FOR EACH ZONE BY TIME PERIOD OBTAINED
20:C*****                  FROM THE DENSITY/PROFILE DATA FILE.
21:C*****
22:      DIMENSION NOMEN(5),DENS(5,7)
23:      INTEGER QUAN(7)
24:100      FORMAT(1X,A6,2X,5A6)
25:101      FORMAT(1X,A6,1X,5A6)
26:200      FORMAT(26X,7I9)
27:201      FORMAT(7I7)
28:300      FORMAT(18X,7F5.1)
29:301      FORMAT(7F5.2)
30:30      READ(7,100,END=99)LIN,(NOMEN(J),J=1,5)
31:      WRITE(3,101)LIN,(NOMEN(J),J=1,5)
32:      READ(8,200,END=99)(QUAN(K),K=1,7)
33:      WRITE(3,201)(QUAN(K),K=1,7)
34:      DO 10 J=1,5
35:      READ(9,300,END=99)(DENS(J,K),K=1,7)
36:      DO 10 L=1,7
37:      DENS(J,L)=DENS(J,L)/100
38:10      CONTINUE
39:      DO 20 M=1,7
40:      WRITE(3,301)(DENS(N,M),N=1,5)
41:20      CONTINUE
42:      GO TO 30
43:99      STOP
44:      END

```

Figure III.6.3

UNCLASSIFIED\*\*\*EXAMPLE DENSITY/PROFILE OUTPUT DATA FROM THE LFA/TAPE UTILITY\*\*\*U

1:	A1 3198	.0	.0	.0	.0	.0	.0
2:	A1 3198	.0	.0	.0	.0	.0	.0
3:	A1 3198	.0	.0	.0	.0	.0	.0
4:	A1 3198	70.5	70.5	60.5	74.5	74.5	74.5
5:	A1 3198	79.5	79.5	39.5	75.4	25.4	25.4
6:	A1 4752	.0	.0	.0	.0	.0	.0
7:	A1 4752	.0	.0	.0	.0	.0	.0
8:	A1 4752	30.7	36.7	36.7	74.3	24.3	24.3
9:	A1 4752	63.3	63.3	63.3	75.7	75.7	65.7
10:	A1 4752	.0	.0	.0	.0	.0	.0
11:	A2 2496	4.9	6.0	1.6	3.7	3.7	3.7
12:	A2 2496	13.4	16.6	20.4	20.9	20.9	20.9
13:	A2 2496	80.0	75.4	66.9	64.5	64.5	64.5
14:	A2 2496	1.5	11.5	11.1	71.0	21.6	21.6
15:	A2 2496	.0	.0	.0	.0	.0	.0
16:	A2 3770	.0	.0	.0	.0	.0	.0
17:	A2 3770	.0	.0	.0	.0	.0	.0
18:	A2 3770	.0	.0	.0	.0	.0	.0
19:	A2 3770	.0	.0	.0	.0	.0	.0
20:	A2 3770	.0	.0	.0	.0	.0	.0
21:	A2 3828	.0	.0	.0	.0	.0	.0
22:	A2 3828	10.0	10.0	10.0	10.0	10.0	10.0
23:	A2 3828	25.0	23.2	28.2	30.8	30.8	30.8
24:	A2 3828	10.1	18.0	16.4	17.6	17.6	17.6
25:	A2 3828	54.9	48.8	45.4	41.5	41.5	41.5
26:	A2 4044	.0	.0	.0	.0	.0	.0
27:	A2 4044	.0	.0	.0	.0	.0	.0
28:	A2 4044	20.0	20.0	20.0	.0	.0	.0
29:	A2 4044	36.2	36.2	36.2	50.0	50.0	50.0
30:	A2 4044	43.6	43.6	43.6	50.0	50.0	50.0
31:	A2 4318	.0	.0	.0	.0	.0	.0
32:	A2 4318	.0	.0	.0	.0	.0	.0
33:	A2 4318	.0	.0	.0	.0	.0	.0
34:	A2 4318	50.0	50.0	60.0	70.0	70.0	80.0
35:	A2 4318	50.0	50.0	40.0	30.0	30.0	20.0
36:	A2 4455	.0	.0	.0	.0	.0	.0
37:	A2 4455	10.0	10.0	10.0	10.0	20.0	30.0
38:	A2 4455	70.0	30.0	30.0	40.0	40.0	50.0
39:	A2 4455	60.0	60.0	60.0	50.0	40.0	20.0
40:	A2 4455	.0	.0	.0	.0	.0	.0
41:	A2 4463	.0	.0	.0	.0	.0	.0
42:	A2 4463	.0	.0	.0	.0	.0	.0
43:	A2 4463	40.5	35.4	39.7	40.7	44.9	44.9
44:	A2 4463	36.4	42.9	44.8	40.2	40.7	40.7
45:	A2 4463	23.1	21.7	15.5	14.2	14.7	14.3
46:	A2 4592	.0	.0	.0	.0	.0	.0
47:	A2 4592	.0	.0	.0	.0	.0	.0
48:	A2 4592	50.5	50.8	52.5	55.4	55.4	55.4
49:	A2 4592	42.7	42.9	41.6	44.3	44.3	44.3
50:	A2 4592	6.9	6.2	5.9	.3	.3	.3
51:	A2 4763	.0	.0	.0	.0	.0	.0
52:	A2 4763	.0	.0	.0	.0	.0	.0
53:	A2 4763	100.0	100.0	100.0	100.0	100.0	100.0
54:	A2 4763	.0	.0	.0	.0	.0	.0
55:	A2 4763	.0	.0	.0	.0	.0	.0
56:	A2 4900	.0	.0	.0	.0	.0	.0
57:	A2 4900	.0	.0	.0	.0	.0	.0

Figure III.6.4

UNCLASSIFIED\*\*\*EXAMPLE OF QUANTITY/PROFILE DATA FROM UTILITY LEA/TAPF\*\*\*UNCLASSY

11: A07198	44	44	44	48	48	48	48
21: A14752	4	6	6	7	7	7	7
31: A22496	5338	5809	3720	4823	4823	4823	4823
41: A27770	0	0	0	0	0	0	0
51: A27628	339	403	458	818	918	518	518
61: A24044	33	13	13	18	18	15	15
71: A24318	4	8	35	57	43	43	67
81: A24455	431	788	918	1544	1280	1272	1272
91: A24463	833	787	943	1184	1500	1500	1200
101: A24292	44	48	83	87	57	57	57
111: A24763	5	5	2	2	2	2	2
121: A24900	854	653	744	349	493	9024	1035
131: A27159	38	58	59	47	49	84	14
141: A32444	3441	5153	2710	3843	3486	3722	3766
151: A32508	540	534	893	677	677	677	677
161: A32564	83	43	986	576	538	235	235
171: A32568	3845	5267	2966	3779	3810	3834	3836
181: A32570	530	440	699	954	924	924	924
191: A34457	4	6	6	7	7	7	7
201: A55293	311	148	198	543	543	243	243
211: A55300	85	73	93	133	133	133	133
221: A55304	4	4	6	6	6	6	6
231: A55704	4	6	6	7	7	7	7
241: A56235	8	8	5	4	6	6	6
251: A56243	3344	4038	8491	7158	7510	7236	7257
261: A56400	58	48	84	56	56	56	56
271: A56937	83	48	75	86	86	86	86
281: A59033	535	279	324	392	404	406	406
291: A77877	84	84	84	84	84	84	84
301: A78014	30	30	30	30	30	30	30
311: A79151	508	523	233	249	280	250	250
321: A90110	937	947	9386	1839	1945	1845	1942
331: A90123	348	564	564	344	344	344	344
341: A90344	0	0	0	0	0	0	0
351: A93943	4	4	4	4	4	4	4
361: B01756	48	87	33	951	151	121	121
371: B07752	84	43	68	68	68	68	68
381: B11795	37	955	184	194	194	194	194
391: B13648	33	44	46	58	68	74	74
401: B36238	8	35	15	12	12	12	12
411: B30923	3	5	5	2	2	2	2
421: B43663	377	514	522	310	327	330	330
431: B51098	84	78	73	915	116	156	126
441: B51099	53	49	80	99	88	91	91
451: B67711	54	38	47	63	61	61	61
461: B67423	314	116	116	150	150	120	120
471: B67492	435	415	412	412	412	412	412
481: B57582	8	8	8	8	8	8	8
491: B83856	389	589	443	1510	1237	1288	1255
501: B94404	480	760	3559	1791	1790	1793	1793
511: C15481	3	33	39	39	39	39	39
521: C27414	558	247	348	823	524	524	524
531: C22058	3	7	34	14	14	14	14
541: C22126	35	14	14	16	16	16	16
551: C22811	54	58	32	32	32	32	32
561: C27017	3	7	34	14	14	14	14
571: C27757	50	52	37	304	107	107	107

Figure III.6.5

UNCLASSIFIED\*\*\*EXAMPLE OF THE LINCDE/LIST DATA FILE\*\*\*UNCLASSIFIED

```

1: A03198 AK VEH M218 GM EQ 12A
2: A14752 ADAP TEST CAMERA L-178
3: A22496 AIMING CIRCLE M2 W/E
4: A23770 AIR COND FL/WNDW 6000B
5: A23828 AIR COND F/WA 9000 BTU
6: A24044 AIR COND 18000 BTU
7: A24318 AIR COND 18000 BTU
8: A24455 AIR COND FK AIR-COOL
9: A24463 AIR COND F/WA 18000 BT
10: A24592 AIR COND 18000 BTU
11: A24763 AIR COND F/WA 36000B TUU
12: A24900 AIR COND 36000 BTU
13: A27189 AIR TRF C F AN/TSQ-97
14: A32444 ALARM CML AGT M11
15: A32508 ALARM CML AGT M12
16: A32564 ALARM CML AGT AUTO
17: A32568 ALARM CML AGT AUTO
18: A32570 ALARM CML AGT AUTO
19: A34457 ALGMY FX MX-8409/AAS 24
20: A41666 RDR SET AN/TPQ-37 LP
21: A55293 ANAL CHG B AN/ASM-137
22: A55300 ANAL CHG B AN/ASM-490
23: A55304 ANAL CHG B AN/GSM-261
24: A55704 ANAL FLT LN AN/ASM-80
25: A56235 ANAL SET LS-89A
26: A56243 ANAL SET ENG PTBL S S
27: A56800 ANAL SPTCM AN/UPM-58
28: A56937 ANAL SPTCM AN/UPM-84
29: A58033 ANAL SPTCM TS-723/U
30: A77877 ANTENNA GRP A GRA-4
31: A79014 ANTENNA GRP A GRA-12
32: A79151 ANTENNA GRP AN/GRA-50
33: A99943 ATTENUVR VAR CN-1035/G
34: B01756 AUGER EARTH SM4A
35: B11795 BOTTLE CLEANG AN/TAM-4
36: B18648 BAKERY PLT M-1945LP
37: B30238 BARGE ASSY SET 5X12
38: B30923 BARGE DECK CGG NP 00
39: B43663 BATH U PTBL GED LP
40: B45597 BTRY CHGR PP72068 Q/U
41: B51098 BEACON SET AN/TRN-30V1
42: B51099 BEACON SET AN/TRN-30V2
43: B63711 BIN STG AGGR PTBL 60 T
44: B67423 BINOCULAR EL AN/PAS-5
45: B67492 BINOCULAR INFRARED
46: B83582 BOAT BRDG ERECT 8D 27
47: B83856 BOAT LAND INFLT 15 MAN
48: B84404 BOAT RECON PNEU 3-MAN
49: C18481 BREAKER PAV-DRILL
50: C20414 BRIDGE ARMD VEH
51: C22058 BRIDGE ERECT SET FIX
52: C22126 BRIDGE ERECT SET FB UK
53: C22811 BRIDGE FXD HI-WAY ALUM
54: C23017 BRIDGE FXD HI-WAY
55: C25757 BRIDGE FLTG RAFT LT
56: C26031 BRIDGE EREC SET
57: C36120 BLDZR EM F/M60 SER TMS

```

Figure III.6.6

UNCLASSIFIED...EXAMPLE OF THE OUTPUT FROM UTILITY ITMID/TEMP...UNCLASSIFIED

```

1: A03198 AK VEH M218 GM EQ P1A
2:      54      64      44      58      78      88      98
3: .00 .00 .00 .80 .20
4: .00 .00 .00 .80 .20
5: .00 .00 .00 .80 .20
6: .00 .00 .00 .80 .20
7: .00 .00 .00 .80 .20
8: .00 .00 .00 .80 .20
9: .00 .00 .00 .75 .25
10: A14752 ADAP TEST CAMERA LM178
11:      10      16      16      17      17      17      17
12: .00 .00 .20 .80 .00
13: .00 .00 .20 .80 .00
14: .00 .00 .20 .80 .00
15: .00 .00 .15 .85 .00
16: .00 .00 .15 .85 .00
17: .00 .00 .35 .65 .00
18: .00 .00 .35 .65 .00
19: A22496 AIMING CIRCLE M2 W/E
20:      6615      6699      8820      8823      8823      8823      8978
21: .25 .25 .50 .00 .00
22: .25 .25 .50 .00 .00
23: .25 .25 .50 .00 .00
24: .25 .25 .50 .00 .00
25: .25 .25 .50 .00 .00
26: .25 .25 .50 .00 .00
27: .25 .25 .50 .00 .00
28: A23770 AIR COND FL/WNDW 60008
29:      0      0      0      0      0      0      0
30: .00 .00 .00 .00 .00
31: .00 .00 .00 .00 .00
32: .00 .00 .00 .00 .00
33: .00 .00 .00 .00 .00
34: .00 .00 .00 .00 .00
35: .00 .00 .00 .00 .00
36: .00 .00 .00 .00 .00
37: A23828 AIR COND F/WA 9000 BTU
38:      889      993      996      998      998      998      998
39: .00 .00 .25 .25 .50
40: .00 .00 .25 .25 .50
41: .00 .00 .25 .25 .50
42: .00 .00 .25 .25 .50
43: .00 .00 .25 .25 .50
44: .00 .00 .25 .25 .50
45: .00 .00 .25 .25 .50
46: A24044 AIR COND 18000 BTU
47:      53      53      53      55      55      55      55
48: .00 .00 .00 .50 .50
49: .00 .00 .00 .50 .50
50: .00 .00 .00 .50 .50
51: .00 .00 .00 .50 .50
52: .00 .00 .00 .30 .70
53: .00 .00 .00 .30 .70
54: .00 .00 .00 .30 .70
55: A24318 AIR COND 18000 BTU
56:      15      25      42      57      63      83      97
57: .00 .00 .25 .75 .00

```

Figure III.6.7

THIS PAGE IS INTENTIONALLY LEFT BLANK

## CHAPTER 7

### UTILITY - ITMID/REC-A

**7.1 DESCRIPTION OF PROCESSING:** The file produced by this utility program is the final ITMID for input to the ELCON program. The source is approximately 54 lines of executable code that consists of logic tests and read and write statements. There are no computations in this program.

**7.1.1 PURPOSE:** The purpose of this program is to merge data from the first produced ITMID/TEMP element with data from the aged ITMID/FINAL element which was produced and used in the previous study. The element should still be cataloged under the previous study's file. The result of this merging will be the ITMID/FINAL element of the current study.

The ITMID/FINAL element is composed of 3 record types:

- o The first record type is the Master Record. It contains information which describes the line item as a whole such as its name, vulnerability class, and supply levels.
- o The second record type denotes the quantity of this item that will be available for each of the seven time periods being played in the study.
- o The third record type describes density profile of this item in each of five zones or areas of the battlefield. There will be seven occurrences of this record for each occurrence of the Master Record.

This current program concentrates on the Master Record. To the new ITMID/FINAL element the program will copy all the data of the new ITMID/TEMP element and only columns 40-54 of the previous study's ITMID/FINAL element. If a new ITMID/TEMP Line Item cannot match an existing line item in the previous ITMID/FINAL, columns 40-54 are blank in the new ITMID/FINAL. The line item is written out to another element "MISSED/ITMID-CODES". Study analysts, using this can then identify those items which need attention on the new ITMID/FINAL element and manually add in the required data.

If line items on the previous ITMID/FINAL have no corresponding items on the ITMID/TEMP, no records will be copied.

The program works using the following assumptions:

All elements are sorted in ascending order using LINCODE.

For each Master Record there will be one corresponding QUANTITY Record and 7 corresponding DENSITY/PROFILE Records.

**7.1.2 PROGRAM INPUT/OUTPUT STRUCTURE:** The overall structure of the utility can be seen in FIGURE III.7.1. The logic flow is pictured in the flow chart in FIGURE III.7.2. Finally the actual source code can be examined in FIGURE III.7.3.

**7.1.2.A INPUT DATA AND DATA BASE:** The utility has two input files. One file is the ITMID/TEMP file that was produced by the previous utility ITMID/TEMP. The other file is the ITMID/FINAL file which was produced during the last study. This file should be cataloged as an element under the previous study's name.

Figure III.7.4 presents the record layouts and an example of data within the ITMID/TEMP file. Figure III.7.5 demonstrates the same for the ITMID/FINAL file from the previous study.

**7.1.2.B OUTPUT DATA AND DATA FILES:** This utility produces one primary element as output and one secondary. The primary product of this utility is the ITMID/FINAL element. It details each item of equipment being played in the scenario, providing quantity and density statistics. The second product of this utility is the MISSED/ITMID-CODES elements. Entries in this element identify those items being played in this exercise which were not present in the previous study. As a result columns 40-54 of the ITMID/FINAL element will be blank. In order for the study to be completed these fields must be normally edited and correct data supplied.

Figure III.7.5 presents the record layouts and example of data for the ITMID/FINAL file. Figure III.7.6 presents the record layout and data example of the MISSED/ITMID-CODES file.

#### **7.1.2.C DATA ELEMENT DICTIONARY:**

The following section will identify and define all variables used in this utility.

<u>NAME</u>	<u>DEFINITION</u>
DENS1	A one dimensional, 5 member array used to read densities from the old ITMID/FINAL file.
DENS2	A one dimensional array of 5 members. Each member denotes the fraction of the total quantity of an item which is allocated to a combat zone during a specific time period. These values will be provided by the current IMIT/TEMP file and will be the values that are written out to the new ITMID/FINAL file.
LIN1	The 6 character line code which uniquely identifies the item of equipment within the system. This particular data element also identifies the record being read from the old ITMID/FINAL file.
LIN2	Identifies the item being read from the ITMID/TEMP file. This will be the variable written out to the new ITMID/FINAL file.

NOMEN	The nomenclature consists of 30 characters which is organized into an array of 5 members, 6 characters each.
QUAN1	An array of 7 members, 7 integers each, which contains the authorized quantity of this item for a specific time period. This variable is used as a dummy in which to place the quantities coming in from the old ITMID/FINAL file.
QUAN2	This variable performs the same function as the QUAN1 variable except that it contains data from the ITMID/TEMP file. This will be the quantity data which will be written out to the new ITMID/FINAL file.
RECA	This variable will contain 15 characters of alphanumeric data. This data will be provided by the old ITMID/FINAL "A" Record for this item. It will contain such data as: historical classification, in-theater shipment by air.
SEQ	A sequence number generated by the utility and assigned to the "A" record of this item of equipment.
I, NN, J, JJ, II, K, KK, L, M, NN, LL	Various integers used as subscripts in the arrays of the utilities.

7.2 OPERATING ENVIRONMENT: This program is implemented on the UNIVAC Executive-8 operating system.

7.2.1 SUPPORT SOFTWARE: This routine requires the FORTRAN IV compiler and the normal UNIVAC 1100/82 system facilities.

7.2.2 I/O DEVICES: This utility will use input and output files that reside or are written to disk. Refer to Volume I for the program execution runstream.

7.3 MAINTENANCE PROCEDURES: There are no explicit procedures developed.

7.3.1 PROGRAMMING CONVENTIONS: This utility follows no unique conventions; FORTRAN programming conventions apply.

7.3.2 INTERNAL ERROR ROUTINES: There are no explicit error handling routines written into this utility. Provisions have been taken, however, to identify items of equipment which were found on the ITMID/TEMP file but not on the previous ITMID/FINAL file. These items are written out to the MISSED/ITMID-CODES file.

# ITMID/REC-A STRUCTURE

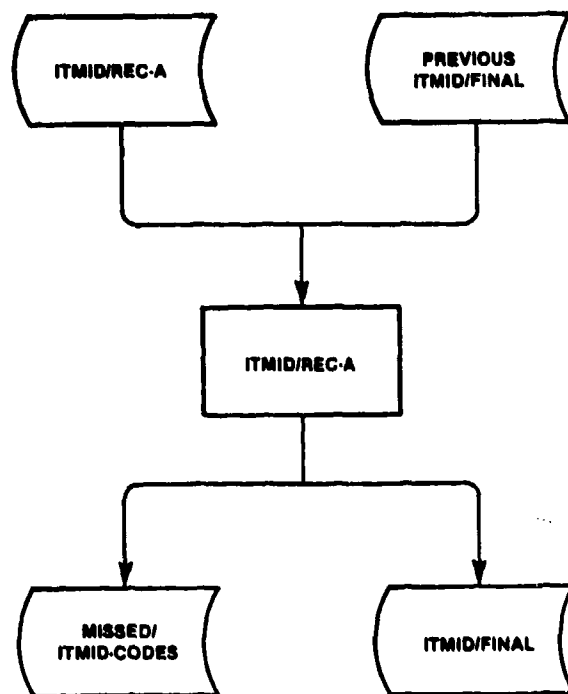


Figure III.7.1

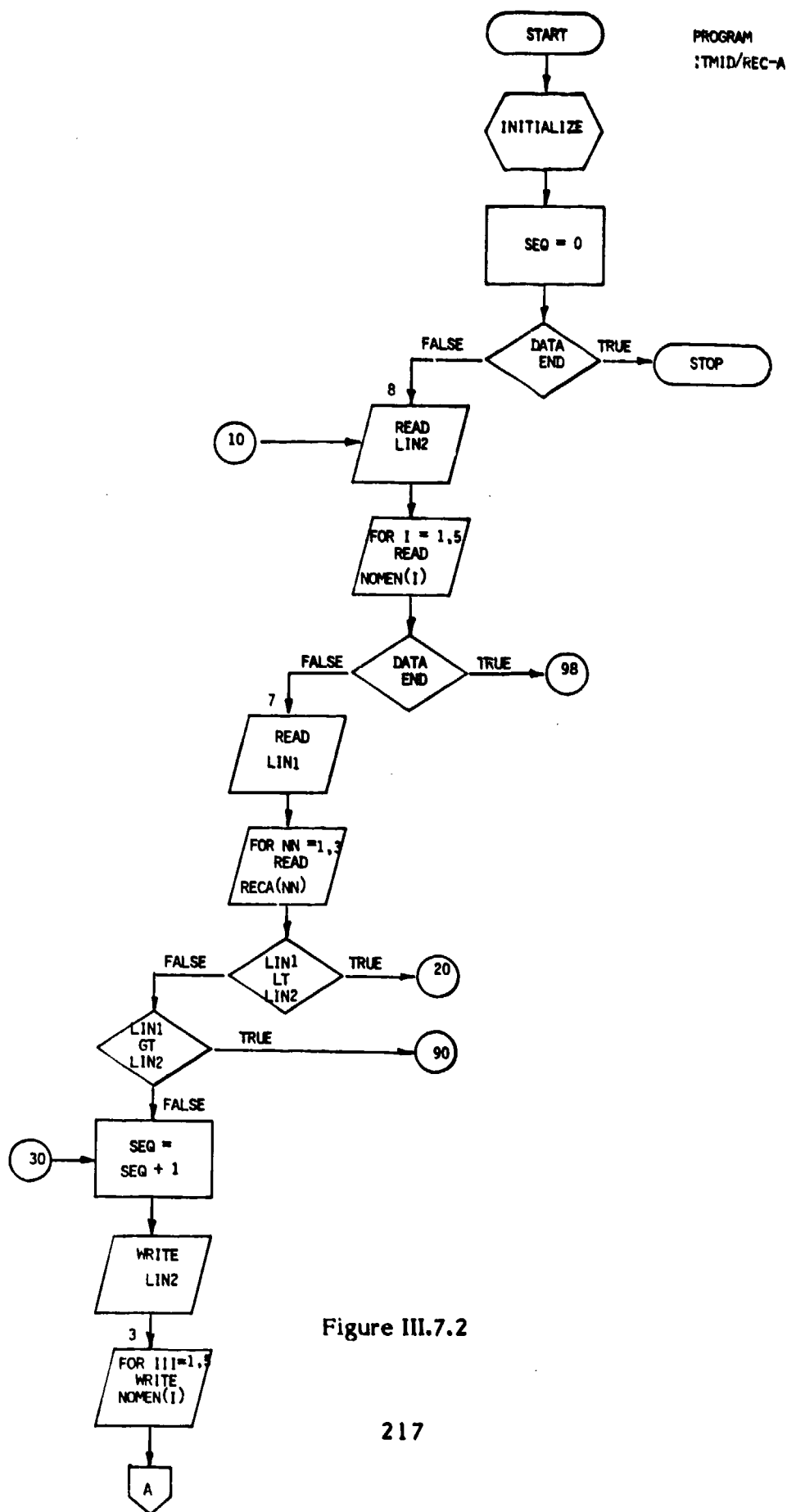


Figure III.7.2

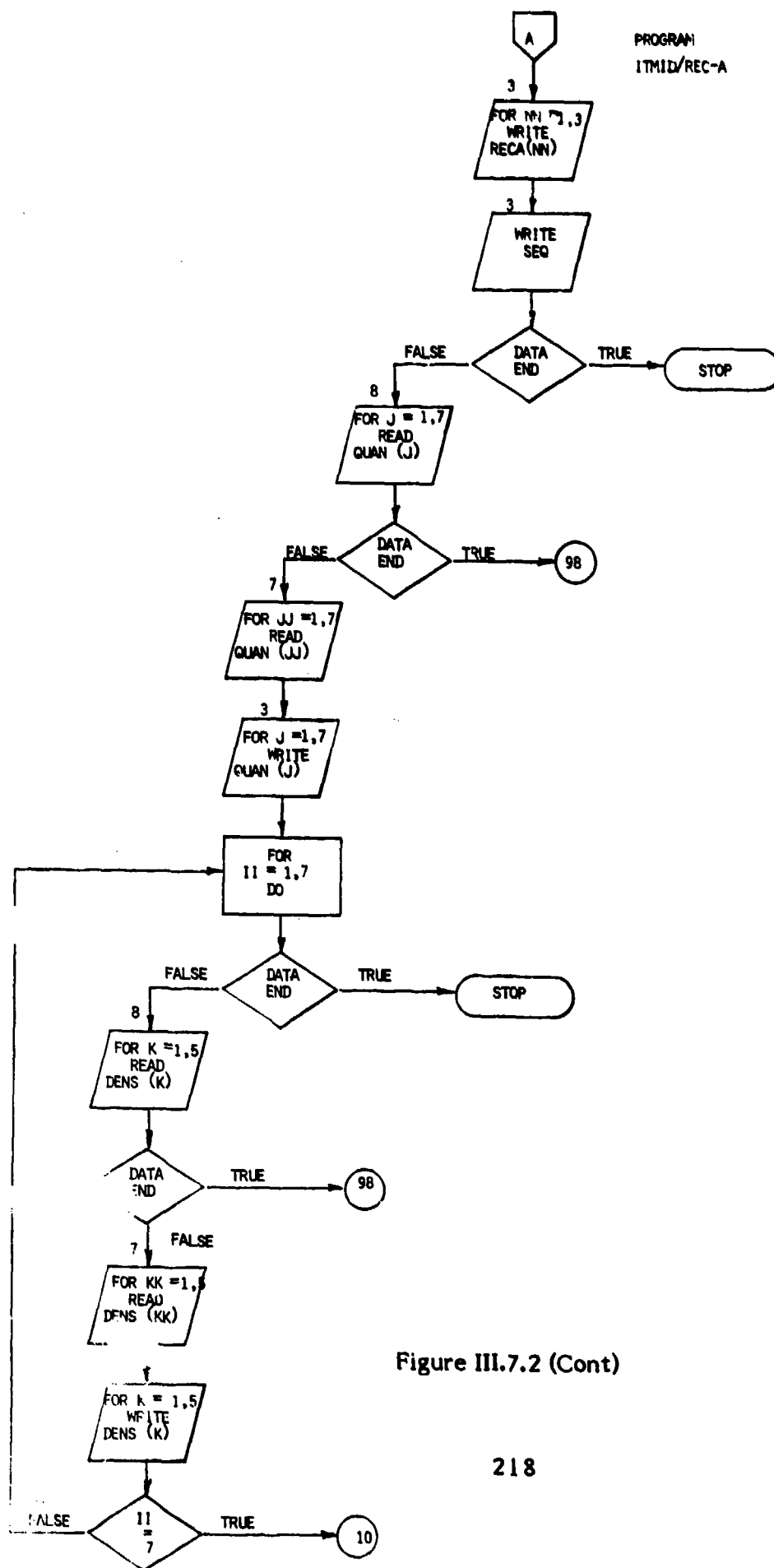


Figure III.7.2 (Cont)

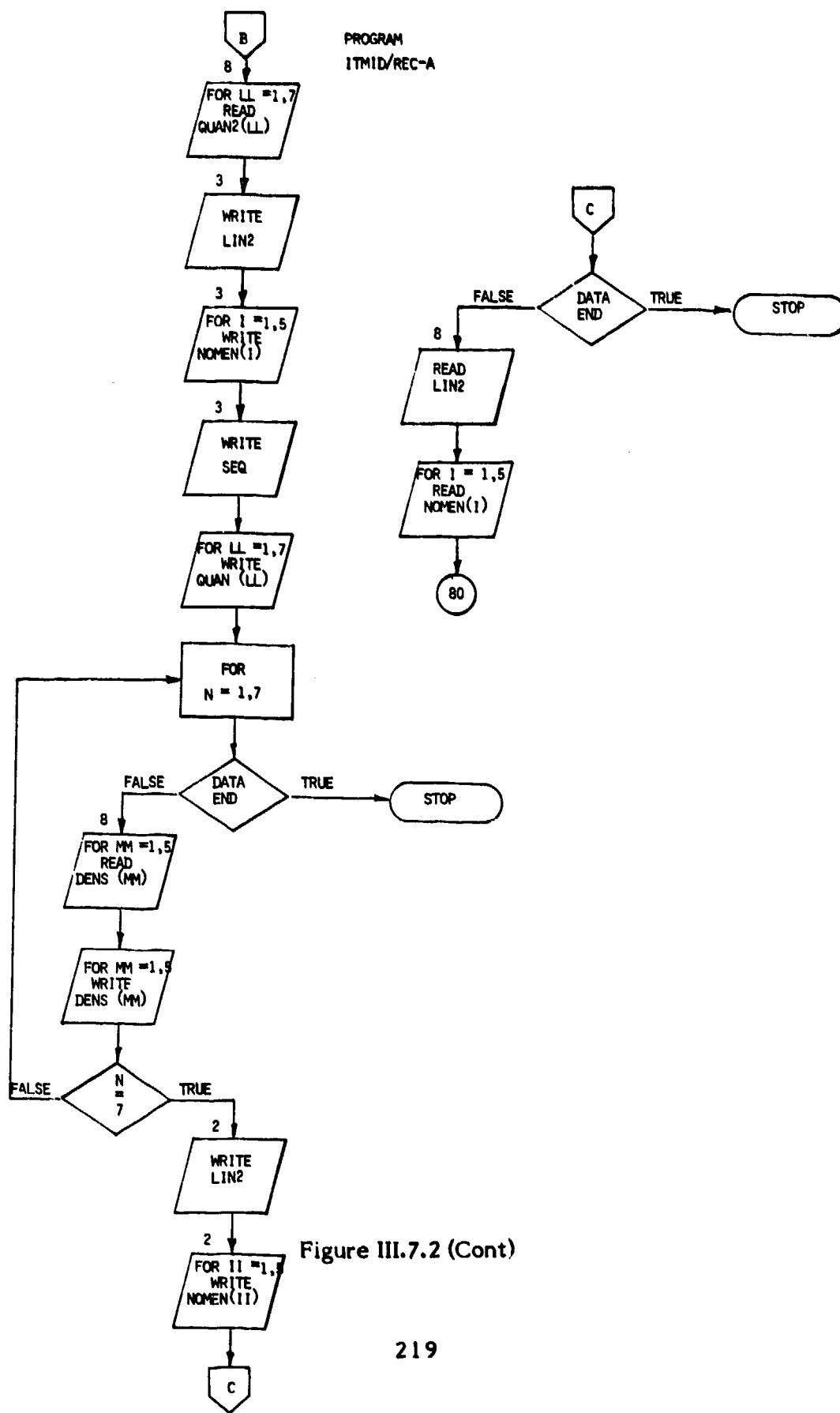
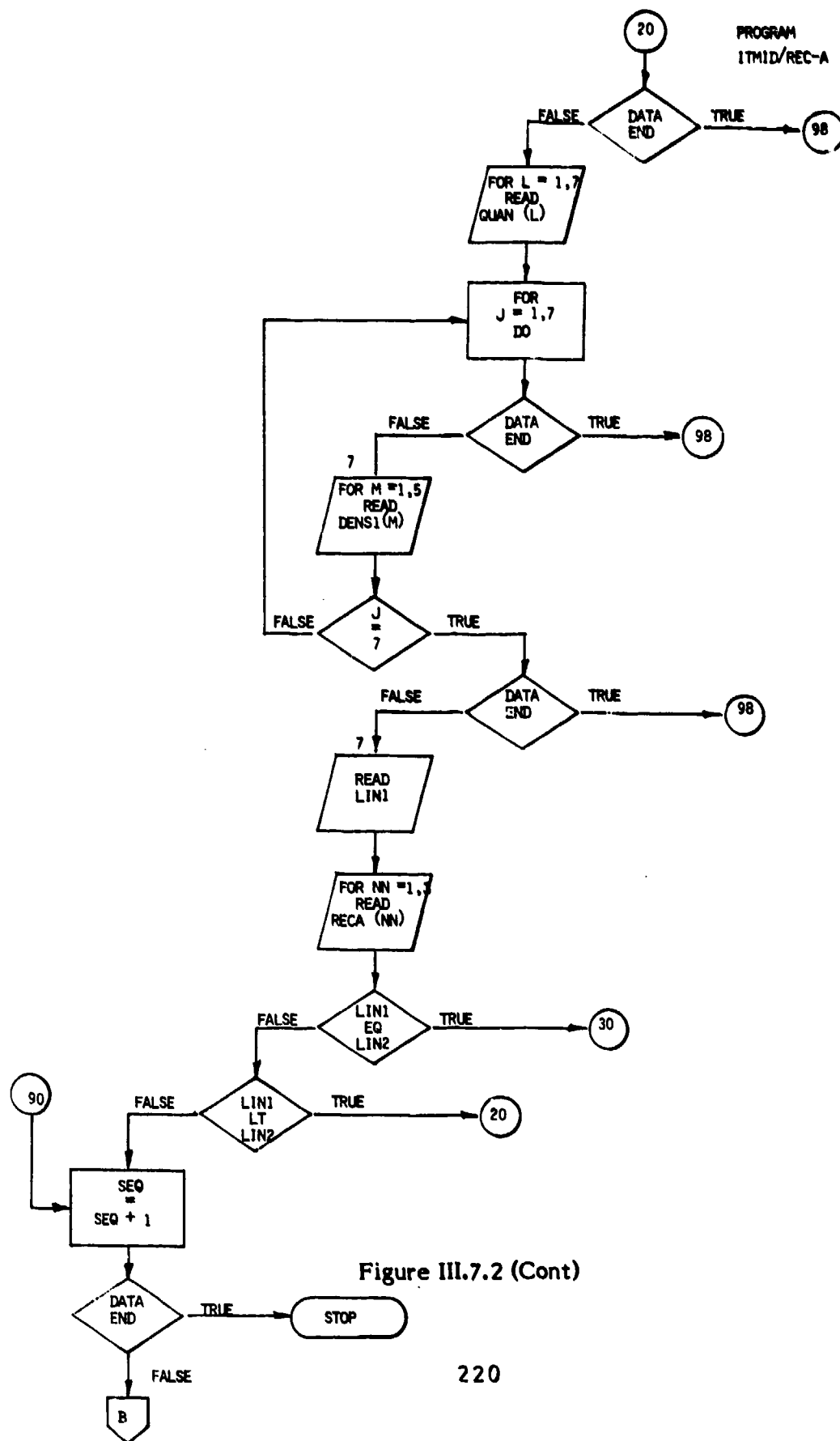


Figure III.7.2 (Cont)



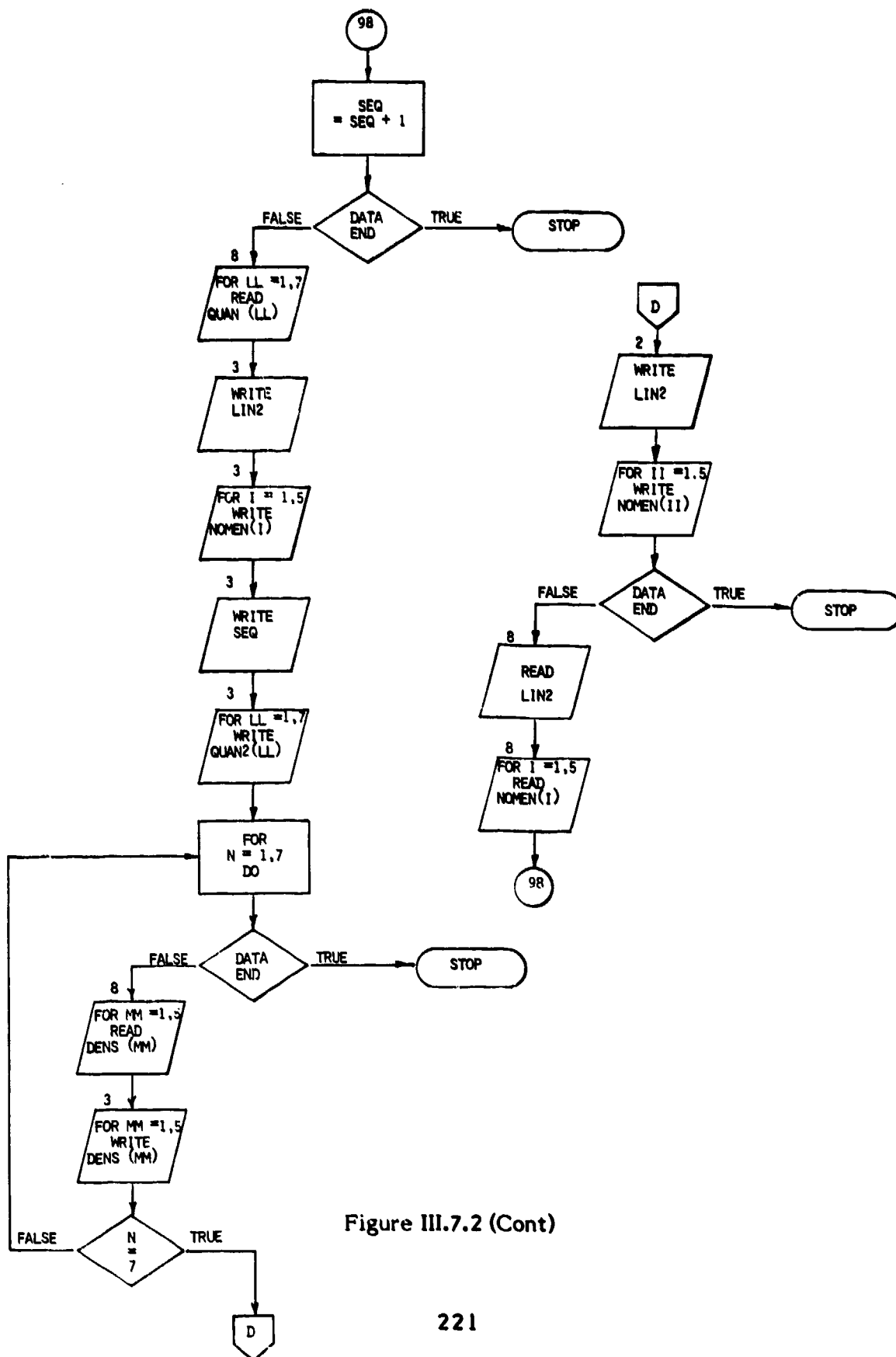


Figure III.7.2 (Cont)

UNCLASSIFIED\*\*\*FILE NAME:82XCT ELEMENT NAME:ITHID/REC-A\*\*\*UNCLASSIFIED

```

1:*****
2:*****
3:*****PURPOSE OF PROGRAM:
4:*****
5:*****THIS UTILITY PROGRAM IS DESIGNED TO EXTRACT DATA FROM A PREVIOUS
6:*****STUDY'S ITHID DATA FILES ON RECORD A COLUMNS 40-54 IAW TABLE
7:*****4-6, CAA-D-79-3 (ELCON) AND ADDS A SEQUENCE NUMBER IN COLUMNS
8:*****55-58 OF THE NEW STUDY'S ITHID FILE. ADDITIONALLY, THE PROGRAM
9:*****PRODUCES A SEPARATE DATA ELEMENT FILE FOR WHICH LIN CODE
10:*****MATCHES WERE NOT FOUND.
11:*****THE ITHID FILE PRODUCED BY THIS UTILITY WILL BE THE FINAL
12:*****ITHID FOR INPUT TO THE ELCON. ONLY MANUAL EDIT CHANGES WILL
13:*****BE MADE TO THE ITHID FILE FROM THIS POINT ON. THE UTILITY
14:*****PROGRAM "82XCT.CATEGORIES" SHOULD BE RUN ON THE FINAL ITHID
15:*****FILE. THIS WILL CREATE A REPORT ON THE CEM, ARTILLERY AND
16:*****HISTORICAL CATEGORIES/CLASSES OF EACH LIN CODE. THE REPORT
17:*****IS SENT TO THE STUDY SPONSOR FOR REVIEW AND APPROVAL.
18:*****
19:*****
20:*****VARIABLE NAME DICTIONARY
21:*****VARIABLE NAME      DEFINITION
22:*****LIN1              LIN CODE FROM OLD ITHID
23:*****LIN2              LIN CODE FROM NEW ITHID
24:*****NOMEN              NOMENCLATURE FROM NEW ITHID
25:*****QUAN1              DUMMY USED TO READ QUANTITIES FROM OLD ITHID
26:*****QUAN2              QUANTITIES FROM NEW ITHID
27:*****DENS1              DUMMY USED TO READ DENSITIES FROM OLD ITHID
28:*****DENS2              DENSITIES FROM NEW ITHID
29:*****SEQ                SEQUENCE NUMBER FOR EACH LIN CODE
30:*****RECA              CONTAINS DATA FROM COLUMNS 40-54
31:*****                  OF RECORD A FROM PREVIOUS ITHID FILE
32:*****
33:*****
34:      DIMENSION NOMEN(5), DENS1(5),DENS2(5),RECA(3)
35:      INTEGER QUAN1(7),QUAN2(7),SEQ
36:100    FORMAT(1X,A6,32X,3A5)
37:200    FORMAT(1X,A6,1X,5A6)
38:300    FORMAT(1X,A6,1X,5A6,1X,3A5,14)
39:400    FORMAT(7I7)
40:450    FORMAT(7I7)
41:500    FORMAT(5F5.2)
42:600    FORMAT(1X,A6,1X,5A6,16X,14)
43:      SEQ=L
44:10    READ(8,200,END=99)LIN2,(NOMEN(I),I=1,5)
45:      READ(7,100,END=96)LIN1,(RECA(NN),NN=1,3)
46:60    IF(LIN1.LT.LIN2)GO TO 20
47:      IF(LIN1.GT.LIN2)GO TO 90
48:30    SEQ=SEQ+1
49:      WRITE(3,300)LIN2,(NOMEN(I),I=1,5),(RECA(NN),NN=1,3),SEQ
50:      READ(1,450,END=99)(QUAN2(J),J=1,7)
51:      READ(7,400,END=98)(QUAN1(JJ),JJ=1,7)
52:      WRITE(3,450)(QUAN2(J),J=1,7)
53:      GO TO 11=1,7
54:      READ(8,500,END=99)(DENS2(K),K=1,5)
55:      READ(7,500,END=98)(DENS1(KK),KK=1,5)
56:      WRITE(3,500)(DENS2(K),K=1,5)
57:50    CONTINUE

```

Figure III.7.3

UNCLASSIFIED\*\*\*FILE NAME:82XDT ELEMENT NAME:ITMID/REC-A\*\*\*UNCLASSIFIED

```

51:      GO TO 10
52:20    READ(7,400,END=96)(QUAN1(L),L=1,7)
60:      DO 70 IJ=1,7
61:      READ(7,500,END=96)(DENS1(M),M=1,5)
62:70    CONTINUE
63:      READ(7,100,END=96)LIN1,(RECA(NN),NN=1,3)
64:      IF(LIN1.EQ.LIN2)GO TO 30
65:      IF(LIN1.LT.LIN2)GO TO 20
66:90    SEQ=SEQ+1
67:      READ(8,400,END=99)(QUAN2(LL),LL=1,7)
68:      WRITE(3,600)LIN2,(NOMEN(I),I=1,5),SEQ
69:      WRITE(3,450)(QUAN2(LL),LL=1,7)
70:      DO 60 N=1,7
71:      READ(8,500,END=99)(DENS2(MM),MM=1,5)
72:      WRITE(3,500)(DENS2(MM),MM=1,5)
73:60    CONTINUE
74:      WRITE(2,200)LIN2,(NOMEN(II),II=1,5)
75:      READ(8,200,END=99)LIN2,(NOMEN(I),I=1,5)
76:      GO TO 60
77:98    SEQ=SEQ+1
78:      READ(8,450,END=99)(QUAN2(LL),LL=1,7)
79:      WRITE(3,600)LIN2,(NOMEN(I),I=1,5),SEQ
80:      WRITE(3,450)(QUAN2(LL),LL=1,7)
81:      GO 40 N=1,7
82:      READ(8,500,END=99)(DENS2(MM),MM=1,5)
83:      WRITE(3,500)(DENS2(MM),MM=1,5)
84:40    CONTINUE
85:      WRITE(2,200)LIN2,(NOMEN(II),II=1,5)
86:      READ(8,200,END=99)LIN2,(NOMEN(I),I=1,5)
87:      GO TO 98
88:99    STOP
89:      END

```

Figure III.7.3 (Cont)

UNCLASSIFIED\*\*\*EXAMPLE OF THE OUTPUT FROM UTILITY IYINQ/TEMP\*\*\*UNCLASSIFIED

```

1: A03198 AK VEH M218 OM EG F1A
2:      54      64      44      58      78      86      98
3: .00 .00 .00 .80 .20
4: .00 .00 .00 .80 .20
5: .00 .00 .00 .80 .20
6: .00 .00 .00 .80 .20
7: .00 .00 .00 .80 .20
8: .00 .00 .00 .80 .20
9: .00 .00 .00 .75 .25
10: A14752 ADAM TEST CAMERA LM178
11:      10      16      16      17      17      17      17
12: .00 .00 .20 .80 .00
13: .00 .00 .20 .80 .00
14: .00 .00 .20 .80 .00
15: .00 .00 .15 .85 .00
16: .00 .00 .15 .85 .00
17: .00 .00 .35 .65 .00
18: .00 .00 .35 .65 .00
19: A22496 AIMING CIRCLE M2 W/E
20:      6615      6699      8820      8823      9823      8823      8978
21: .25 .25 .50 .00 .00
22: .25 .25 .50 .00 .00
23: .25 .25 .50 .00 .00
24: .25 .25 .50 .00 .00
25: .25 .25 .50 .00 .00
26: .25 .25 .50 .00 .00
27: .25 .25 .50 .00 .00
28: A23770 AIR COND FL/WNDW 60008
29:      0      0      0      0      0      0      0
30: .00 .00 .00 .00 .00
31: .00 .00 .00 .00 .00
32: .00 .00 .00 .00 .00
33: .00 .00 .00 .00 .00
34: .00 .00 .00 .00 .00
35: .00 .00 .00 .00 .00
36: .00 .00 .00 .00 .00
37: A23828 AIR COND F/WA 9000 BTU
38:      889      993      998      998      998      998      998
39: .00 .00 .25 .25 .50
40: .00 .00 .25 .25 .50
41: .00 .00 .25 .25 .50
42: .00 .00 .25 .25 .50
43: .00 .00 .25 .25 .50
44: .00 .00 .25 .25 .50
45: .00 .00 .25 .25 .50
46: A24044 AIR COND 15000 BTU
47:      53      53      53      55      55      55      55
48: .00 .00 .00 .50 .50
49: .00 .00 .00 .50 .50
50: .00 .00 .00 .50 .50
51: .00 .00 .00 .50 .50
52: .00 .00 .00 .30 .70
53: .00 .00 .00 .30 .70
54: .00 .00 .00 .30 .70
55: A24318 AIR COND 18000 BTU
56:      15      25      42      57      63      83      97
57: .00 .00 .25 .75 .00

```

Figure III.7.4

UNCLASSIFIED\*\*\*EXAMPLE OF THE OUTPUT OF UTILITY ITMID/REC-1\*\*\*UNCLASSIFIED

1:	AD3198	AK	VEH	M218	GM	EQ	P1A			0	522	30	0	1	2	1
2:	54		64		44		58	78		88		88				
3:	.00	.00	.00	.80	.20											
4:	.00	.00	.00	.80	.20											
5:	.00	.00	.00	.80	.20											
6:	.00	.00	.00	.80	.20											
7:	.00	.00	.00	.80	.20											
8:	.00	.00	.00	.80	.20											
9:	.00	.00	.00	.75	.25											
10:	A14752	ADAP	TEST	CAMERA	LM178					01636	30	0	1	2	2	
11:	10		16		16	17		17		17		17				
12:	.00	.00	.20	.80	.00											
13:	.00	.00	.20	.80	.00											
14:	.00	.00	.20	.80	.00											
15:	.00	.00	.15	.85	.00											
16:	.00	.00	.15	.85	.00											
17:	.00	.00	.35	.65	.00											
18:	.00	.00	.35	.65	.00											
19:	A22496	AIMING	CIRCLE	M2	W/E					01636	30	0	1	2	3	
20:	6615		6699		8823	8823		8823		8923		8978				
21:	.25	.25	.50	.00	.00											
22:	.25	.25	.50	.00	.00											
23:	.25	.25	.50	.00	.00											
24:	.25	.25	.50	.00	.00											
25:	.25	.25	.50	.00	.00											
26:	.25	.25	.50	.00	.00											
27:	.25	.25	.50	.00	.00											
28:	A23770	AIR	COND	FL/WNDW	60008					01833	30	0	1	2	4	
29:	0		0		0	0		0		0		0				
30:	.00	.00	.00	.00	.00											
31:	.00	.00	.00	.00	.00											
32:	.00	.00	.00	.00	.00											
33:	.00	.00	.00	.00	.00											
34:	.00	.00	.00	.00	.00											
35:	.00	.00	.00	.00	.00											
36:	.00	.00	.00	.00	.00											
37:	A23828	AIR	COND	F/WA	9000	BTU				01833	30	0	1	2	5	
38:	889		993		998	998	998	998		998		998				
39:	.00	.00	.25	.25	.50											
40:	.00	.00	.25	.25	.50											
41:	.00	.00	.25	.25	.50											
42:	.00	.00	.25	.25	.50											
43:	.00	.00	.25	.25	.50											
44:	.00	.00	.25	.25	.50											
45:	.00	.00	.25	.25	.50											
46:	A24044	AIR	COND	18000	BTU					01833	30	0	1	2	6	
47:	53		53		53	55		55		55		55				
48:	.00	.00	.00	.50	.50											
49:	.00	.00	.00	.50	.50											
50:	.00	.00	.00	.50	.50											
51:	.00	.00	.00	.50	.50											
52:	.00	.00	.00	.30	.70											
53:	.00	.00	.00	.30	.70											
54:	.00	.00	.00	.30	.70											
55:	A24318	AIR	COND	18000	BTU					01833	30	0	1	2	7	
56:	15		25		42	57		63		83		97				
57:	.00	.00	.25	.75	.00											

Figure III.7.5

UNCLASSIFIED\*\*\*EXAMPLE OF OUTPUT DATA MISSED/ITMID-CODES FROM UTILITY ITMID/TEMP

1: A90118 ARMT SS HEL 7.62MM M23  
2: A90123 ARMT SS HEL 7.62MM M24  
3: A90344 ARMT SS HEL 7.62MM M41  
4: J88047 INSTL KIT ELEC EQ: MK15 2/GSG-  
5: L45016 LCHR RKT ACFT M158A1  
6: L45063 LCHR RKT ACFT M200A1  
7: L52040 LENS CONE A R LA-370A  
8: L52041 LENS CONE A R LA-371A  
9: L52042 LENS CONE A R LA-372A  
10: L91701 MG CAL.50 HVY FIXED  
11: L92260 MG 7.62MM ACFT LT  
12: L95939 MAINT A K MK-1192/ARM  
13: M11621 MASK CBR PROT ACFT M24  
14: Z00570 AERIAL RADIAC AN/ADR-6  
15: Z21489 DETECT SET AN/APR39V2  
16: Z43971 MICNS AIR DATA TML (RPV)  
17: Z43999 MISSION PAYLOAD NIGHT (PV)  
18: Z44712 MOUNT GM LCHR DRAGON  
19: Z50159 POSITION LOC RPT ABN U  
20: Z62820 AIR VEHICLE (RPV)  
21: Z66150 MISSION PAYLOAD DAYLIGH (RPV)  
22: Z93241 TRK CGO 10T MLRS RESUPP Y

Figure III.7.6

## CHAPTER 8

### UTILITY - TOE/1ST RUN

**8.1 DESCRIPTION OF PROCESSING:** This program performs only logic-tested read and write functions; no computations are performed.

**8.1.1 PURPOSE/FUNCTIONS:** The objective of this utility is to extract from the Table of Equipment (TOE) Master File obtained from FASTALS, authorized LINCOCES and quantities by unit type and place them into the output file TOE/1ST RUN. This function is accomplished by comparing the nine character Standard Requirements Codes (SRC) of the TOE Master File with the SRC's of the Arrayed Units file, which is generated by the study. When a match is detected the appropriate information is extracted from the TOE Master File and written to the output file.

SRC's in the TOE Master File for which no match could be made in the ARRAYED Units file are written to a second output file named "UNMATCHED/UNITS.

The newly created TOE/1ST RUN file will be used as input to the following utility, SCRUB/TOE.

**8.1.2 PROGRAM INPUT/OUTPUT STRUCTURE:** The overall structure of the utility is pictured in FIGURE III.8.1. The logic flow found in the utility is described by the flow chart in FIGURE III.8.2. The source code for this utility is pictured in FIGURE III.8.3.

**8.1.2.A INPUT DATA AND DATA BASE:** This utility uses two data sets as input files. The first file is the FASTALS TOE Master File which is obtained from the support Forces Group, Forces Analyses Directorate, USACAA. The file is a copy of the FASTALS TOE data for the outyear of the current WARF study. The second input data set is the ARRAYED/UNITS file which is normally created manually using the system editor and data from the current study's program file. Both files are sorted in ascending order based upon the 9 character SRC code.

Figure III.8.4 depicts the record layout and sample data for the FASTALS TOE Master File. Figure III.8.5 depicts the record layout and sample data for the ARRAYED/UNITS file.

**8.1.2.B OUTPUT DATA AND DATA FILES:** - This utility produces two data sets as output. The first, UNMATCHED/UNITS, is simply a list of units for which an SRC match with the TOE Master File could not be made. the second file, TOE/1st RUN, is a compilation of all units for which an SRC match was made and the list of major items of equipment which is authorized for the unit. The file has two major record types. The first record type, of which, there will be one for each second record type, of which there can be one or more, denotes each item of equipment, identified by its' LINCOCES, authorized to the unit and quantity of items. This file will be used as input to the following utility SCRUB/TOE.

Figure III.8.6 depicts the record layout for the UNMATCHED/UNITS file and an example of the file. Figure III.8.7 displays the record layout for the TOE/1st RUN file and present an example of the data that can be found in the file.

### 8.1.2.C DATA ELEMENT DICTIONARY:

The following section will identify and define all variables used in this utility.

<u>NAME</u>	<u>DEFINITION</u>
LIN	The six character alphanumeric code which uniquely identifies an item of equipment.
ISRC	The nine character Standard Requirements Code found in the ARRAYED/UNITS file. The utility will establish this variable as an array.
JSRC	The Standard Requirement Code found in the TOE MASTER file.
IUNIT	A 25 character alphanumeric field which describes the combat unit being analyzed. This information is supplied by the ARRAYED/UNITS file. The utility establishes this file as an array.
ATEST	A 1 character field used by the utility to analyze the records in the TOE MASTER file. The TOE MASTER file consists of 2 record types. The first record type is identified by an "A" in the first character position; the second record type is flagged by a "B" in the first character position.
IQTY	A 4 character integer field which denotes the quantity authorized of a specific item of equipment. This data is supplied by the TOE MASTER file.
UNID	A 4 character alphanumeric code which uniquely identifies a unit within the study. This data is supplied by the ARRAYED/UNITS file. The utility dimensions this variable as an array.
I, II, J	Various integer variables used as subscripts to the arrays used in the utility.

8.2 OPERATING ENVIRONMENT: The program is implemented on the EXECUTIVE-8 operating system.

8.2.1 SUPPORT SOFTWARE: This routine requires the ASCII FORTRAN compiler and the UNIVAC 1100/82 system facilities.

8.2.2 I/O DEVICES: This utility will use input files which reside on disk and will in turn produce files which will reside on disk.

8.3 MAINTENANCE PROCEDURES: There are no explicit maintenance procedures developed.

8.3.1 PROGRAMMING CONVENTIONS: This utility takes advantage of the structuring capabilities of ASCII FORTRAN. Unlike FORTRAN IV, the programmer can format its logic using IF-THEN-ELSE Structures. Each level of logic may be indented 2 spaces in order to show the structure and logic.

8.3.2 INTERNAL ERROR ROUTINES: There are no specific error routines written into the utility. However the utility is structured so that if it reaches the end of the ARRAYED/UNITS file before all the equipment items and units in the FASTAL TOE Master File have been processed it will branch to and print an appropriate error message before it terminates.

# TOE1STRUN STRUCTURE

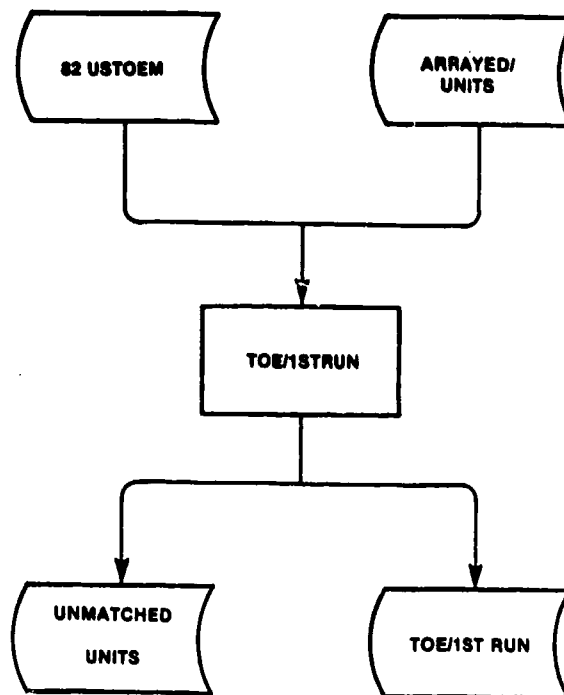


Figure III.8.1

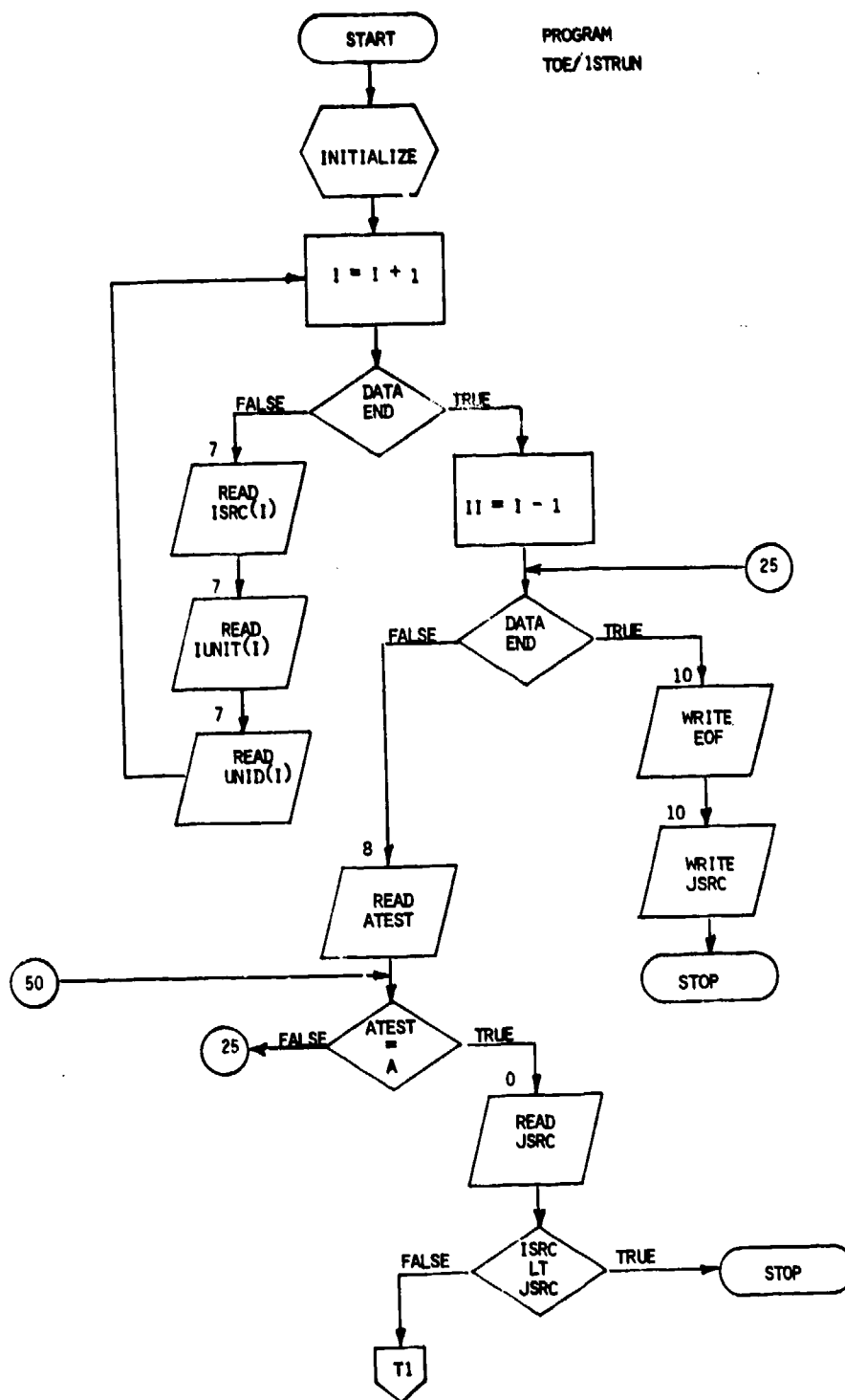


Figure III.8.2

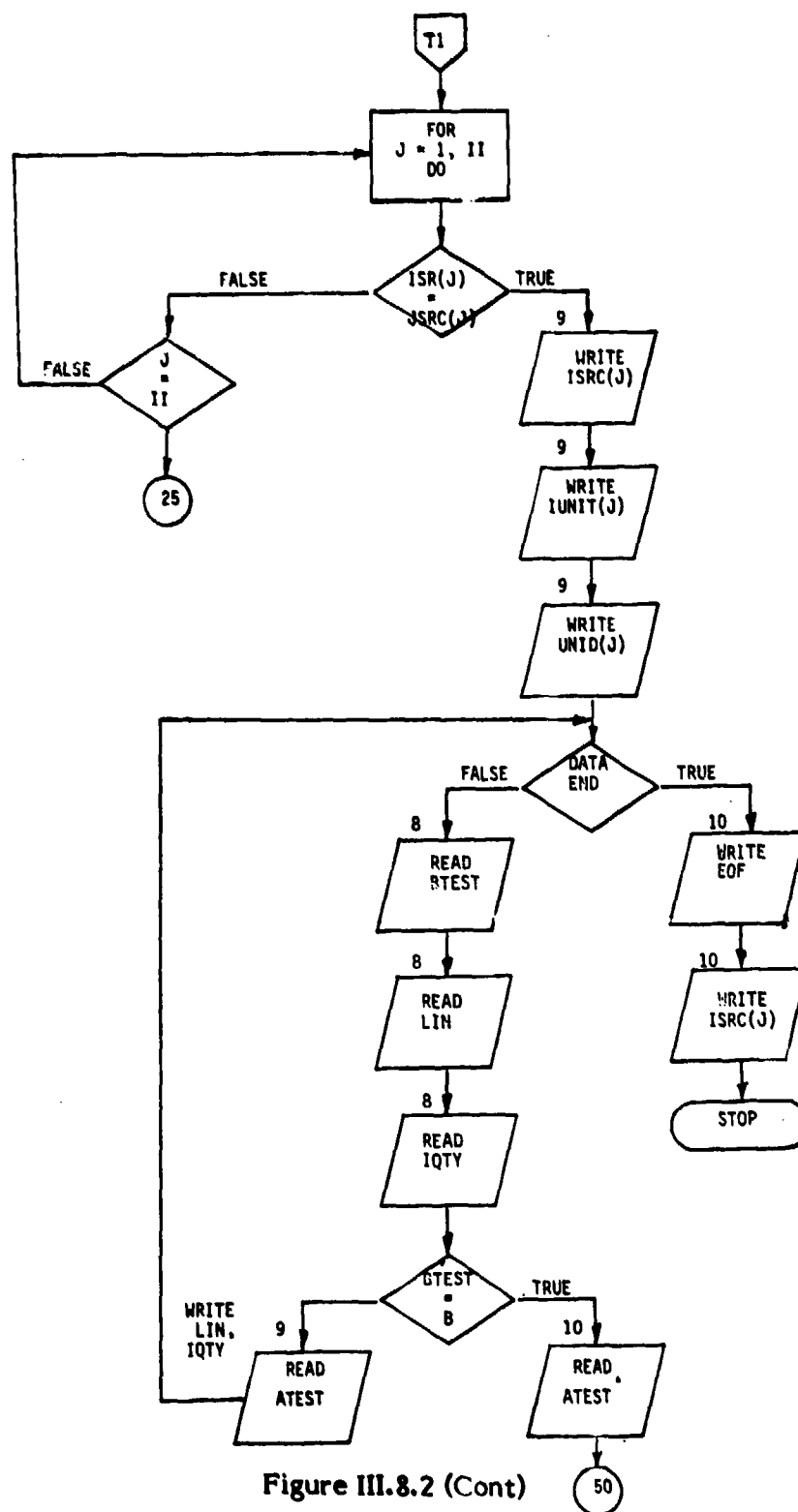


Figure III.8.2 (Cont)

UNCLASSIFIED\*\*\*FILE NAME:82XQT ELEMENT NAME:TOE/1STRUN\*\*\*UNCLASSIFIED

```

1:C*****
2:C*****
3:C*****THIS UTILITY PROGRAM HAS BEEN WRITTEN IN ASCII FORTRAN.
4:C*****
5:C*****
6:C*****THIS UTILITY IS DESIGNED TO MATCH THE FASTAL TOE MASTER FILE
7:C*****WITH THE TYPE UNITS LIST BY SRC AND EXTRACT FROM THE TOE MASTER
8:C*****FILE AUTHORIZED LIN CODES AND QUANTITIES BY UNIT TYPE. INPUT FILES TO
9:C*****THIS UTILITY INCLUDE THE FASTALS TOE MASTER FILE AND STUDY GENERATED
10:C*****TYPE UNITS LIST BY SRC FILE.
11:C*****THE TYPE UNIT LIST BY SRC IS A MANUALLY EDITED ELEMENT
12:C*****"ARRAYED/UNITS" OF THE CURRENT STUDY'S PROGRAM FILE.
13:C*****THE CONTENT OF THIS LIST INCLUDES SRC, TITLE AND ARRAY CODE
14:C*****FOR EACH TYPE OF UNIT ARRAYED IN THE FOUR STYLIZED POSTURE ARRAYS.
15:C*****THE OUTPUT OF THIS UTILITY WILL SERVE AS INPUT TO THE UTILITY
16:C*****"82XQT.SCRUB/TOE".
17:C*****
18:C*****
19:C*****VARIABLE NAME DICTIONARY
20:C*****VARIABLE NAME      DEFINITION
21:C*****ISRC(I)            STANDARD REQUIREMENTS CODE/TOE#
22:C*****                    FROM ARRAYED UNITS' DATA FILE
23:C*****IUNIT(I)           NOMENCLATURE OF TYPE UNIT
24:C*****LIN                LIN CODE FROM THE MASTER TOE FILE
25:C*****IGTY               QUANTITY AUTHORIZED EACH LIN
26:C*****IUNID(I)           TYPE UNIT NUMBER
27:C*****JSRC               STANDARD REQUIREMENTS CODE/TOE#
28:C*****                    FROM THE TOE MASTER FILE(FASTALS)
29:C*****BTEST              "B" TYPE RECORD (LIN CODE & GYT RECORD)
30:C*****ATEST              "A" TYPE RECORD (SRC/TOE #)
31:C*****
32:C*****
33:      CHARACTER*4 UNID(200)
34:      CHARACTER*9 ISRC(200)
35:      CHARACTER*25 IUNIT(200)
36:      CHARACTER*6 LIN
37:      CHARACTER*9 JSRC
38:      CHARACTER*1 ATEST
39:      CHARACTER*1 BTEST
40:10      I=I+1
41:      READ(7,200,END=20)ISRC(I),IUNIT(I),UNID(I)
42:200    FORMAT(1X,A9,2X,A25,A4)
43:      GO TO 10
44:20      II=I-1
45:25      READ(8,300,END=95)ATEST
46:300    FORMAT(A1)
47:50      IF(ATEST.EQ.'A')
48:      *THEN
49:          READ(10,301)JSRC
50:301      FORMAT(1X,A9)
51:          IF(ISRC(II).LT.JSRC)GO TO 99
52:          DO 30 J=1,II
53:              IF(ISRC(J).EQ.JSRC)
54:              * THEN
55:                  WRITE(9,400)ISRC(J),IUNIT(J),UNID(J)
56:400      FORMAT(A9,5X,A25,6X,A4)
57:40      READ(8,302,END=97)BTEST,LIN,IGTY

```

Figure III.8.3

UNCLASSIFIED\*\*\*FILE NAME:82XQT ELEMENT NAME:TOE/1STRUN\*\*\*UNCLASSIFIED

```

59:302      FORMAT(A1,A6,I4)
59:      IF(RTEST.EQ.'B')
60:      *      THEN
61:      WRITE(9,401)LIN,I6TY
62:401      FORMAT(5X,A6,2X,I4)
63:      GO TO 40
64:      ELSE
65:      READ(10,300)ATEST
66:      GO TO 50
67:      ENDIF
68:      ENDF
69:30      CONTINUE
70:      GO TO 25
71:      ELSE
72:      GO TO 25
73:      ENDF
74:95      WRITE(10,505)
75:505      FORMAT('REACH END OF FILE ON END=95')
76:      WRITE(10,506)JSRC
77:506      FORMAT(A9)
78:      GO TO 99
79:97      WRITE(10,502)
80:502      FORMAT('REACH END OF FILE ON END=97')
81:      WRITE(10,506)ISRC(J)
82:99      STOP
83:      END

```

Figure III.8.3 (Cont)

UNCLASSIFIED\*\*\*EXAMPLE OF DATA FILE 02USTOEM (FASTALS MASTER TOC)\*\*\*UNCLASSIFIED

1:A010660900  
2:B A0 32100001  
3:B A3 09460002  
4:B A3 20600002  
5:B A5 52930001  
6:B A7 22600002  
7:B A9 01180004  
8:B B0 71260001  
9:B B6 77660002  
10:B C5 26010001  
11:B C5 30120001  
12:B C6 87190012  
13:B C6 88560004  
14:B C6 42270001  
15:B C8 47750001  
16:B C8 62130001  
17:B C8 91450110  
18:B C8 92130110  
19:B D7 94810001  
20:B D8 01160001  
21:B D9 05380001  
22:B D9 88880001  
23:B D9 90250001  
24:B E0 05330002  
25:B E1 08350001  
26:B E2 42810001  
27:B E3 30830001  
28:B E4 57660003  
29:B E4 58200005  
30:B E5 86010002  
31:B E7 00640001  
32:B E7 02010001  
33:B E7 08170001  
34:B E8 45310001  
35:B G4 45690001  
36:B G8 52020001  
37:B H0 23000001  
38:B H8 38170002  
39:B J4 29760001  
40:B J4 39180003  
41:B J4 61100002  
42:B J4 76170001  
43:B J4 90550001  
44:B J4 93980002  
45:B J5 43300001  
46:B K2 38140001  
47:B K2 53420008  
48:B K3 10420002  
49:B K3 17950004  
50:B K8 72430001  
51:B K8 72690001  
52:B K8 72730001  
53:B K9 73190001  
54:B K9 82510001  
55:B L0 09840001  
56:B L1 05320004  
57:B L4 45950006

Figure III.8.4

UNCLASSIFIED\*\*\*EXAMPLE OF ARRAYED/UNITS DATA FILE\*\*\*UNCLASSIFIED

1:	03087H700	NBC DEF CO	CH01
2:	05145H710	CRT ENG BN	EN02
3:	05146H710	HMC ENG BN	EN03
4:	05147H000	ENG CO	EN04
5:	05148H710	BRG CO	EN05
6:	06302H000	HMB DIVARTY	FA06
7:	06365H000	155MM SP BN	FA07
8:	06366H000	HMB 155MM BN	FA08
9:	06367H000	155MM PTRY	FA09
10:	06369H000	SVC BTRY 155MM	FA10
11:	06395B110	8"/GSRS BN	FA11
12:	06396B110	HMB 8"/GSRS BN	FA12
13:	06397B000	8"PTRY	FA13
14:	06398B100	GSRS BTRY	FA14
15:	06399B000	SVC BTRY 8"/GSRS	FA15
16:	06445H100	8" SP BN(CORPS)	FA16
17:	06446H100	HMB 8" BN(CORPS)	FA17
18:	06447H100	8" BTRY(CORPS)	FA18
19:	06449H100	SVC BTRY 8" (CORPS)	FA19
20:	06515B000	GSRS BN(CORPS)	FA20
21:	06516B000	HMB GSRS BN(CORPS)	FA21
22:	06517B000	GSRS BTRY(CORPS)	FA22
23:	07045H020	MECH INF BN	ME23
24:	07046H010	HMC MECH INF BN	ME24
25:	07047H010	MECH INF CO	ME25
26:	07047H9X9	MECH INF PLT	ME26
27:	07048H020	CS CO INF	ME27
28:	08035H000	MED BN	MD28
29:	08036H000	HMC MED BN	MD29
30:	08037H000	MED CO	MD30
31:	08123H000	CRT SUPT HOSRITAL (CORPS)	MD31
32:	09127H410	MED AMB CO(CORPS)	MD32
33:	09137H200	MED AIR AMB CO(CORPS)	MD33
34:	09038H300	ORD CO CONV AMMO(CORPS)	OR34
35:	09047H400	SPEC AMMO DS CO(CORPS)	OR35
36:	09048G800	SPEC AMMO DS/GS CC(CORPS)	OR36
37:	09268H800	MNT BTRY HAWK(CORPS)	AD37
38:	09557H510	MISSILE SUPT CO	AD38
39:	10007H000	SBS CO	QM39
40:	10207H300	PETRL PL&TML OP CC(CORPS)	QM40
41:	11035H000	SIGNAL BN	SC41
42:	11036H000	HMC SIG BN	SC42
43:	11037H000	CMD OPS CO	SC43
44:	11038H000	FWD COMM CO	SC44
45:	11039H000	SIG SUPT OP CO	SC45
46:	12017H610	AG CO	AG46
47:	14037H610	FIN CO	FC47
48:	17004H000	HMC ARMD DIV	AR48
49:	17035H010	TANK 105MM BN	AR49
50:	17036H000	HMC TANK BN	AR50
51:	17037H010	TANK 105MM CO	AR51
52:	17037H9X9	TANK 105MM PLT	AR52
53:	17039H000	CS CO TANK	AR53
54:	17042H000	HMC ARMD BDE	CA54
55:	17105H020	ARMD CAV SQDN	CA55
56:	17106H000	HMT CAV SQDN	CA56
57:	17108H000	AIR CAV TRP	CA57

Figure III.8.5

UNCLASSIFIED\*\*\*EXAMPLE OF OUTPUT FROM UTILITY TOF/15\*PUN\*\*\*UNCLASSIFIED

1:REACHED END OF FILE ON END=95  
2:12255H700

Figure III.8.6

UNCLASSIFIED\*\*\*EXAMPLE OF OUTPUT FROM UTILITY TOE/15\*RUN\*\*\*UNCLASSIFIED

		NBC DEF CO	CH01
1:	03087H700		
2:	A32316	1	
3:	A32444	9	
4:	A32540	9	
5:	A72260	1	
6:	B49272	111	
7:	B67766	10	
8:	C52601	1	
9:	C53012	1	
10:	C53149	2	
11:	C89145	83	
12:	C89213	83	
13:	E00533	12	
14:	E45820	1	
15:	E70064	1	
16:	F61880	9	
17:	J43918	11	
18:	J46110	1	
19:	K87243	9	
20:	K87269	4	
21:	K87392	1	
22:	K87393	1	
23:	K87456	1	
24:	L44595	10	
25:	L63994	10	
26:	L92386	13	
27:	M11895	111	
28:	M75714	13	
29:	M80002	1	
30:	N54691	1	
31:	N96741	1	
32:	P43177	2	
33:	P95592	9	
34:	Q19339	18	
35:	Q19681	1	
36:	Q20935	38	
37:	Q21483	19	
38:	Q53001	4	
39:	Q54174	4	
40:	Q56783	6	
41:	Q78282	1	
42:	R73791	9	
43:	R94977	110	
44:	U01305	2	
45:	V15018	9	
46:	V31211	6	
47:	W32593	1	
48:	W32867	1	
49:	W33004	7	
50:	W51910	1	
51:	W95400	13	
52:	W95811	11	
53:	X39447	1	
54:	X40146	11	
55:	X40968	9	
56:	X58367	1	
57:	X60833	13	

Figure III.8.7

THIS PAGE IS INTENTIONALLY LEFT BLANK

THIS PAGE IS INTENTIONALLY LEFT BLANK

## CHAPTER 9

### UTILITY - SCRUB/TOE

9.1 DESCRIPTION OF PROCESSING: This program performs only logic tested read and write functions; no computations are performed.

9.1.1 PURPOSE/FUNCTIONS: The purpose of this utility is to eliminate all major items of equipment from the TOE/1st RUN which are not being played in this study. The resulting file is referred to as the SCRUBBED/TOE. To accomplish this function the TOE/1st RUN utility was produced by the TOE/1st RUN utility and passed against the LINCONE/LIST file which was created manually earlier from the hard copy list of LINCONEs provided by the study's sponsor. Only those items in the TOE/1st RUN file for which a match can be made in the LINCONE/LIST file will be copied into the SCRUBBED/TOE file and will continue in the study.

9.1.2 PROGRAM INPUT/OUTPUT STRUCTURE: - The overall structure of the utility is pictured in FIGURE III.9.1. The logic followed by the utility is pictured in FIGURE III.9.2. The source code for the utility is listed in FIGURE III.9.3.

9.1.2.A INPUT DATA AND DATA BASE: The SCRUB/TOE uses two files as input data files. One file is the TOE/1st RUN file which was created during the TOE/1st RUN utility which must be completed prior to the execution of this utility. This file identifies the units which are being analyzed under this current study. In addition, each unit contains detailed items of equipment which has been assigned to the unit and the respective quantities for each item. Figure III.9.4 presents the file layout and examples of the data found in the file.

The second file used as input to this utility is the LINCONE/LIST file. This file is created manually using the hard copy list of LINCONEs of the equipment items which are being analyzed in this study. This list of LINCONEs is provided by the study's sponsor. Figure III.9.5 present the record layout and examples of the data found in the file.

9.1.2.B OUTPUT DATA AND DATA FILES: This utility produces one file as output, the SCRUBBED/TOE file. It will be cataloged under the current studies general file, SECRET\*82WARFP86. This file uses the same record layout as the TOE/1st RUN input file to this utility. Further, while this new file contains all the units as the input file, it has eliminated from it all items of equipment which are not being analyzed in this study. If the item is not in the LINCONE/LIST file, it will not be in the resulting SCRUB/TOE file.

Figure III.9.6 depicts the record layout for the file and an example of the data that can be found in it. Once again the record layouts for the SCRUB/TOE and TOE/1st RUN are identical.

### 9.1.2.C DATA ELEMENT DICTIONARY:

The following section identifies and defines all variables used in this utility.

<u>NAME</u>	<u>DEFINITION</u>
ISRC	Standard Requirement Code/TOE number. This 9 character alphanumeric field is supplied by the TOE/-ISTRUN file.
IUNIT	Unit Nomenclature. This is a 25 character alphanumeric field which describes the combat unit. This data is supplied by the TOE/ISTRUN file.
UNID	Unit Identifier. This is a 4 character alphanumeric field which uniquely identifies the unit in the study. This data is supplied by the TOE/ISTRUN file.
LIN1	The 6 character item line code from the TOE MASTER file.
LIN2	The 6 character item line code from the LINC CODE/-LIST file.
IQTY	The authorized quantity for this item as specified in the TOE/ISTRUN file.
ITEST	A 2 character variable which is used to examine columns 48 and 49 of the TOE/ISTRUN record. If these columns are blank (i.e., ITEST=0) it indicates that a record which contains equipment and quantity data is being processed. If ITEST is not blank it indicates a header or unit identification record is being processed.
NLIN	An integer variable which is used to denote the total number of items of equipment is being played in the study as indicated by the LINC CODE/-LIST file. It is also used in the utility as the upper limit of a DO LOOP.
I	An integer variable used as a counter to determine the number of different items of equipment being played in the study.
J	An integer variable used as a subscript.

9.2 OPERATING ENVIRONMENT: This program is implemented on the EXECUTIVE-8 operating system.

9.2.1 SUPPORT SOFTWARE: This routine requires the ASCII FORTRAN compiler and the UNIVAC 1100/82 system facilities.

9.2.2 I/O DEVICES: This utility will use input files which reside on disk and will in turn produce a file which will reside on disk.

9.3 MAINTENANCE PROCEDURES: There are no developed unique or special maintenance procedures or applications.

9.3.1 PROGRAMMING CONVENTIONS: This utility takes advantage of the formatting capabilities of ASCII FORTRAN by using the IF-THEN-ELSE structure. No indenting scheme is used to illustrate the logic.

9.3.2 INTERNAL ERROR ROUTINES: There are no explicit error routines built into this utility. Errors will be detected by the system and must be corrected using available system and ASCII FORTRAN documentation.

# SCRUB/TOE STRUCTURE

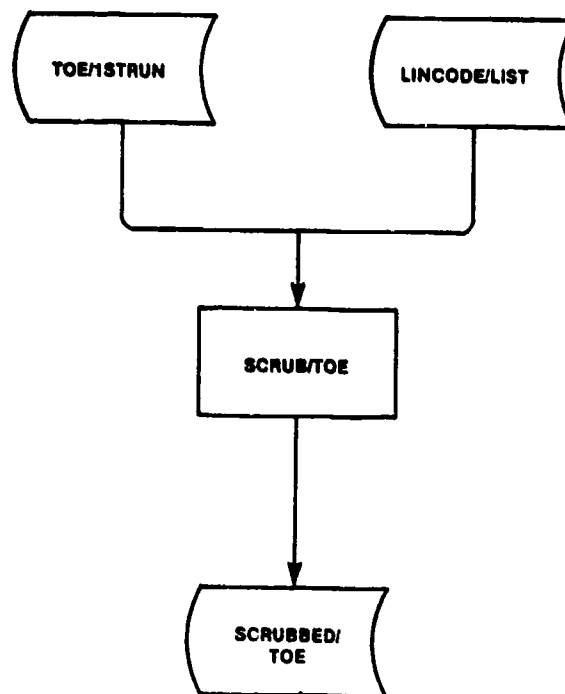


Figure III.9.1

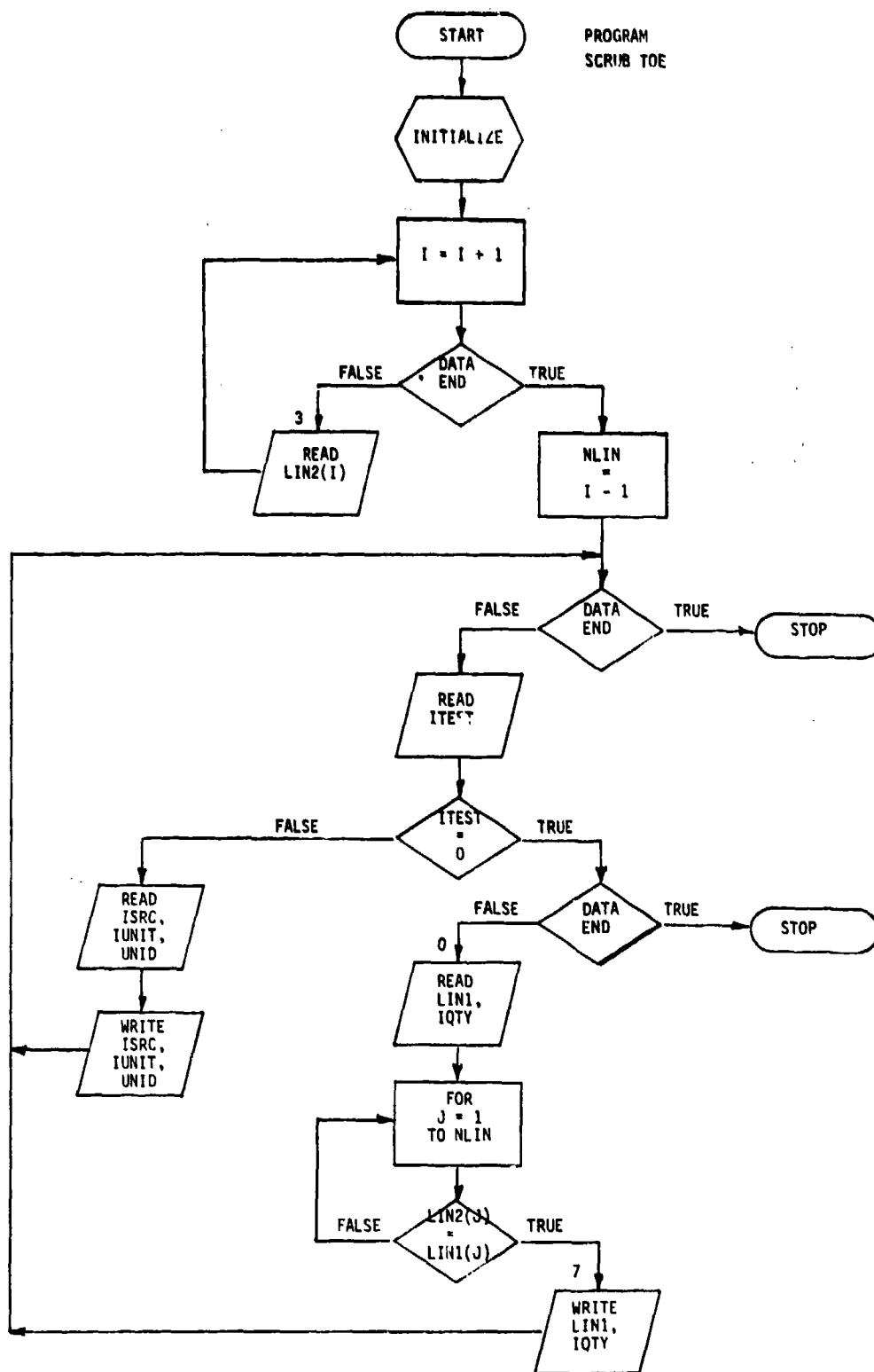


Figure III.9.2

UNCLASSIFIED\*\*\*FILE NAME:82XQT ELEMENT NAME:SCRUB/TOE\*\*\*UNCLASSIFIED

```

1:C*****
2:C*****
3:C*****THIS UTILITY PROGRAM WAS WRITTEN IN ASCII FORTRAN.
4:C*****
5:C*****
6:C*****THIS UTILITY IS DESIGNED TO TAKE THE OUTPUT OF UTILITY "82XQT.TOE/
7:C*****1STRUN" AND DELETE ALL LIN CODES FROM THE VARIOUS TYPE OF UNIT
8:C*****TOE'S WHICH ARE NOT BEING CONSIDERED BY THE WARF STUDY. THE OUT-
9:C*****PUT OF THIS UTILITY WILL BE FURTHER UPDATED BY THE UTILITY "82XQT.
10:C*****TOE/ADD-PLTS AND MANUAL EDIT.
11:C*****
12:C*****
13:C*****VARIABLE NAME DICTIONARY
14:C*****VARIABLE NAME      DEFINITION
15:C*****ISRC              STANDARD REQUIREMENTS CODE/TOE#
16:C*****IUNIT            NOMENCLATURE OF TYPE UNIT
17:C*****LIN1             LINS FROM MASTER TOE FILE
18:C*****LIN2             LINS FROM MARKLIST FOR CURRENT STUDY
19:C*****IQTY             QTY FROM MASTER TOE FILE
20:C*****IDNUM            TYPE UNIT NUMBER
21:C*****ITEST            DUMMY
22:C*****
23:C*****
24:      CHARACTER*9 ISRC
25:      CHARACTER*4 UNID
26:      CHARACTER*31 IUNIT
27:      CHARACTER*6 LIN1
28:      CHARACTER*6 LIN2(2000)
29:10      I=I+1
30:      READ(3,300,END=20)LIN2(I)
31:300    FORMAT(1X,A6)
32:      GO TO 10
33:20      NLIN=I-1
34:30      READ(2,200,END=99)ITEST
35:200    FORMAT(47X,I2)
36:      IF(ITEST.EQ.0)
37:      *THEN
38:      READ(0,201,END=99)LIN1,IQTY
39:201    FORMAT(5X,A6,2X,I4)
40:      DO 40 J=1,NLIN
41:      IF(LIN2(J).EQ.LIN1)
42:      *THEN
43:      WRITE(7,201)LIN1,IQTY
44:      GO TO 45
45:      ENDIF
46:40      CONTINUE
47:45      CONTINUE
48:      ELSE
49:      READ(0,202)ISRC,IUNIT,UNID
50:202    FORMAT(A9,5X,A31,A4)
51:      WRITE(7,202)ISRC,IUNIT,UNID
52:      ENDIF
53:      GO TO 30
54:99      STOP
55:      END

```

Figure III.9.3

UNCLASSIFIED\*\*\*EXAMPLE OF OUTPUT FROM UTILITY TOE/15\*RUN\*\*\*UNCLASSIFIED

	1:03U87H700	NRC DEF CO	CH01
2:	A32316	1	
3:	A32444	9	
4:	A32540	9	
5:	A72260	1	
6:	B49272	111	
7:	B67766	10	
8:	C52601	1	
9:	C53012	1	
10:	C53149	2	
11:	C83145	83	
12:	C89213	83	
13:	E00533	12	
14:	E45820	1	
15:	E70064	1	
16:	F81880	9	
17:	J43518	11	
18:	J46110	1	
19:	K87243	9	
20:	K87269	4	
21:	K87392	1	
22:	K87393	1	
23:	K87456	1	
24:	L44595	10	
25:	L63994	10	
26:	L92386	13	
27:	M11895	111	
28:	M75714	13	
29:	M80002	1	
30:	N54691	1	
31:	N96741	1	
32:	P43177	2	
33:	P95592	9	
34:	Q19339	18	
35:	Q19681	1	
36:	Q20935	38	
37:	Q21483	19	
38:	Q53001	4	
39:	Q54174	4	
40:	Q56783	6	
41:	Q78282	1	
42:	R73791	9	
43:	R94977	110	
44:	U01305	2	
45:	V15018	9	
46:	V31211	6	
47:	W32593	1	
48:	W32867	1	
49:	W33004	7	
50:	W51910	1	
51:	W95400	13	
52:	W95811	11	
53:	X39447	1	
54:	X40146	11	
55:	X40968	9	
56:	X58367	1	
57:	X00633	13	

Figure III.9.4

UNCLASSIFIED\*\*\*EXAMPLE OF THE LINCDE/LIST DATA FILE\*\*\*UNCLASSIFIED

```

1: A03198 AK VEH M218 GM EO P14
2: A14752 ADAP TEST CAMERA LM178
3: A22496 AIMING CIRCLE M2 W/E
4: A23770 AIR COND FL/WNDW 6000B
5: A23328 AIR COND F/WA 9000 BTU
6: A24044 AIR COND 13000 BTU
7: A24318 AIR COND 18000 BTU
8: A24455 AIR COND FV AIR-COOL
9: A24463 AIR COND F/WA 18000 BT
10: A24592 AIR COND 19000 BTU
11: A24763 AIR COND F/WA 36000BTU
12: A24900 AIR COND 36000 BTU
13: A27159 AIR TRF C F AN/TSQ-97
14: A32444 ALARM CML AGT M11
15: A32508 ALARM CML AGT M12
16: A32564 ALARM CML AGT AUTO
17: A32568 ALARM CML AGT AUTO
18: A32570 ALARM CML AGT AUTO
19: A34457 ALGHT FX MX-8409/4AS24
20: A41666 RDR SET AN/TPQ-37 LP
21: A55293 ANAL CHG B AN/ASM-137
22: A55300 ANAL CHG B AN/ASM-490
23: A55304 ANAL CHG B AN/CSM-261
24: A55704 ANAL FLT LN AN/ASM-80
25: A56235 ANAL SET LS-89A
26: A56243 ANAL SET ENG PTBL S S
27: A56800 ANAL SPTCH AN/UPM-58
28: A56937 ANAL SPTCH AN/UPM-84
29: A58033 ANAL SPTCH TS-723/U
30: A77877 ANTENNA GRP AN/GRA-4
31: A79014 ANTENNA GRP AN/GRA-12
32: A79151 ANTENNA GRP AN/GRA-50
33: A99943 ATTENUATR VAR CN-1035/E
34: B01756 AUGER TATH SM4A
35: B11795 BOTTLE CLEAN AN/TAM-4
36: B18648 BAKERY FLT M-1945LP
37: B30230 BARCE ASSY SET 5X12
38: B30923 BARCE DECK CGO NP OC
39: B43663 BATH U PTBL GED LP
40: B45597 BTRY CHCR PF72862 G/U
41: B51098 BEACON SET AN/TRN-30V1
42: B51099 BEACON SET AN/TRN-30V2
43: B63711 BIN STG AGGR PTBL 60 T
44: B67423 BINOCULAR EL AN/PAS-5
45: B67492 BINOCULAR INFRARED
46: B83582 BOAT BRDG ERECT GD 27
47: B83856 BOAT LAND INFLT 15 MAN
48: B84404 BOAT RECON PNEU 3-MAN
49: C18481 BREAKER PAV-DRILL
50: C20414 BRIDGE ARMD VEH
51: C22058 BRIDGE ERECT SET FIX
52: C22120 BRIDGE ERECT SET FB UK
53: C22911 BRIDGE FXD HI-WAY ALUM
54: C23017 BRIDGE FXD HI-WAY
55: C25757 BRIDGE FLTG RAFT LT
56: C25031 BRIDGE EREC SET
57: C3F120 BLDGR EM F/M60 SER TMS

```

Figure III.9.5

UNCLASSIFIED\*\*\*EXAMPLE OF OUTPUT OF UTILITY SCRUB/TOE\*\*\*UNCLASSIFIED

		NRC DEF CO	CH01
1:	03087H700		
2:	A32444	9	
3:	E00533	12	
4:	E45820	1	
5:	E70064	1	
6:	F81880	9	
7:	J43918	11	
8:	J46110	1	
9:	L44595	10	
10:	M75714	13	
11:	N96741	1	
12:	P43177	2	
13:	P95592	9	
14:	Q19339	18	
15:	Q20935	38	
16:	Q21483	19	
17:	Q53001	4	
18:	Q54174	4	
19:	Q56783	6	
20:	Q78282	1	
21:	R94977	110	
22:	U01305	2	
23:	V15018	9	
24:	V31211	6	
25:	W32593	1	
26:	W95400	13	
27:	W95811	11	
28:	X39447	1	
29:	X40146	11	
30:	X40968	9	
31:	X58367	1	
32:	X60833	13	
33:	X63299	1	
34:	05145H710	CBT ENG BN	EN02
35:	A32444	10	
36:	B93582	2	
37:	B83856	18	
38:	B84404	27	
39:	C20414	6	
40:	C25757	2	
41:	C85494	3	
42:	C86213	1	
43:	D11538	3	
44:	D12087	40	
45:	E00533	24	
46:	E45820	8	
47:	E56578	8	
48:	E69242	1	
49:	E70064	5	
50:	E70886	1	
51:	E73626	2	
52:	F39378	3	
53:	F81880	1	
54:	G02204	48	
55:	G02341	50	
56:	H02300	2	
57:	H38787	2	

Figure III.9.6

THIS PAGE IS INTENTIONALLY LEFT BLANK

## CHAPTER 10

### UTILITY - TOE/ADD-PLTS

**10.1 DESCRIPTION OF PROCESSING:** This program performs the computation of dividing a units assets by 3 and rounding to create platoon size elements. All other functions are logic tested read and write statements.

**10.1.1 PURPOSE/FUNCTIONS:** The purpose of this utility is to identify those units in the SCRUBBED/TOE file which are in fact platoon size units rather than company size units and correspondingly divide the quantity issued the unit for each authorized item of equipment by 3. Note: Should the quantity of platoons per company size unit be changed, the program must be updated, and re-compiled.

A second file is also used as input to this utility. This file contains those units which have been flagged as platoon size units. Using this file the utility is able to identify those platoon size units in the SCRUBBED/TOE file, make the necessary adjustments in the quantities of equipment authorized, if necessary and write to the output file (FINAL/TOE) the unit identification and the equipment description and quantity data. The study analyst is responsible for the creating of the PLATOONS file. This FINAL/TOE file is the sole output of this utility and is used as input to the following RAM/MATRIX utility (after any additional manual editing required via the system editor).

**10.1.2 PROGRAM INPUT/OUTPUT STRUCTURE:** The overall structure of the utility is pictured in FIGURE III.8.1. The logic followed by the utility is pictured in FIGURE III.10.2. The source code for the utility is listed in FIGURE III.10.3.

**10.1.2.A INPUT AND DATA BASE:** The TOE/ADD-PLTS utility uses two data files as its input. One file is the SCRUBBED/TOE which was created as a result of the SCRUB/TOE utility. The second file is the PLATOONS file which is manually created using the system online editor. Both files are cataloged under the study's general file; in this instance SECRET\*82WARFP88.

The SCRUBBED/TOE file is a collection of all the units participating in the study accompanied by a detailed list of all items of equipment assigned the unit and the quantity of the item authorized. The file uses two record formats. The first record format is used to describe the particular unit and occurs once per unit. It contains such data as the units Standard Requirement Code, nomenclature of the type of unit, and the type unit ID number. The second record type is used to describe the items of equipment assigned to the unit. This record type will occur as many times per unit (i.e., first record type) as there are items of equipment assigned to the unit and being played in the study. This record consists of two fields. The first field identifies the item of equipment using its LINCODE. The second field denotes the number of individual units of this item that are authorized for this unit. Record layouts and examples of the data in the file can be seen in Figures III.10.4 and III.10.5.

The PLATOONS file contains data which identifies parent units of company size and its corresponding platoon size unit. This file consists of one record type which describes the unit by its Standard Requirement Code, unit nomenclature and unit

ID. The utility assumes the records will be organized in the file so that each parent unit record (company) will be followed immediately by one and only one sub-unit (PLATOON) record. The record layout and examples of data within the file can be seen in Figures III.10.6 and III.10.7.

**10.1.2.B OUTPUT DATA AND DATA FILES:** This utility produces one output file, ITMID/FINAL. The file element is cataloged under the study's general file, in this case SECRET\*82WARFP88. This file contains, in addition to all the data on the company size units, that was found on the SCRUBBED/TOE file; i.e., unit identification data and equipment identification data and authorized quantities, information on platoons which are subordinate to selected parent units. These parent units and subordinate units were identified by the study analysts and transmitted to the utility through the PLATOONS input file. The file will have the same file format structure as the SCRUBBED/TOE input file in that it will have two record types. The first record type will describe the particular unit or subunit and will occur once per unit. The second record will describe the items of equipment assigned the unit and the quantity of each item authorized. This record type will occur once for each item of equipment assigned the unit. Figures III.10.8 and III.10.9 present the file layout for the file and examples of the data that can be found in it.

#### **10.1.2.C DATA ELEMENT DICTIONARY:**

The following section identifies and defines all variables used in this utility.

<u>NAME</u>	<u>DEFINITION</u>
IQTY	This is a 4 character integer variable which denotes the quantity of an item which has been authorized for a particular unit.
IXQTY	This is a 4 character integer variable which holds the quantity of the item assigned to a platoon. This value will be 1/3 of the IQTY value rounded up. This will be a value derived in the utility.
SRC1	The Standard Requirement Code. This data is supplied by the PLATOONS input file. It shows the title of units missing TOE data on platoon size unit.
SRC2	The Standard Requirements Code for the unit from the SCRUBBED/TOE input file.
SRC3	A temporary holding variable into which the first 5 characters of a record from the SCRUBBED/TOE file are read for testing. It should be noted that SRC1 and SRC2 are 9 characters long, whereas SRC3 is 5 characters long.
LIN	The 6 character Line Code of a particular item of equipment. This data is provided by the SCRUBBED/-TOE file.

XLIN	This 6 character variable will hold the line code of the item of equipment being assigned to a unit. This variable is dimensioned as an array by the utility. It is assigned the value of the line code of the item of equipment being processed.
XNOM1	This is a 25 character, alphanumeric variable which contains the title of the combat unit being analyzed. This data comes from the PLATOONS file.
XNOM2	Same as XNOM1. This data comes from the SCRUBBED/TOE file.
UNID1	The 4 character, alphanumeric variable which contains the Unit ID and uniquely identifies this unit within the system. This data is provided by the PLATOONS file.
UNID2	Same as UNID1. This data is provided by the SCRUBBED/TOE file.
I	An integer variable which is used to count the number of types of equipment assigned to a specific unit.
II	An integer variable. This field will contain the total number of types of equipment assigned to a unit. It is similar to the variable "I" above except that I is incremented before the next record is read. Thus, when the last record is read, "I" must be decreased by 1. It is this value (i.e., I-1) that is assigned to "II". It is also used as the upper limit of DO LOOPS.
J	An integer variable used as the subscript of several arrays.

10.2 OPERATING ENVIRONMENT: This program is implemented on the EXECUTIVE-8 operating system.

10.2.2 SUPPORT SOFTWARE : This routine requires the ASCII FORTRAN compiler and the UNIVAC 1100/22 system facilities.

10.2.2 I/O DEVICES : This utility will use input files which reside on disk and will in turn produce a file which will also reside on disk. Refer to Volume I for the execution runstream.

10.3 MAINTENANCE PROCEDURES: There are no developed special maintenance procedures.

10.3.1 PROGRAMMING CONVENTIONS This utility takes advantage of the

formatting capabilities of ASCII FORTRAN. It makes full use of the IF-THEN-ELSE structuring and further enhance the readability of the program by indenting those sections which are nested.

10.3.2 INTERNAL ERROR ROUTINES : There are no explicit error handling routines in this utility. Therefore only errors detected by the system during execution will be identified; the system error message and system documentation must be used to correct the error condition.

# **TOE/ADD-PLTS STRUCTURE**

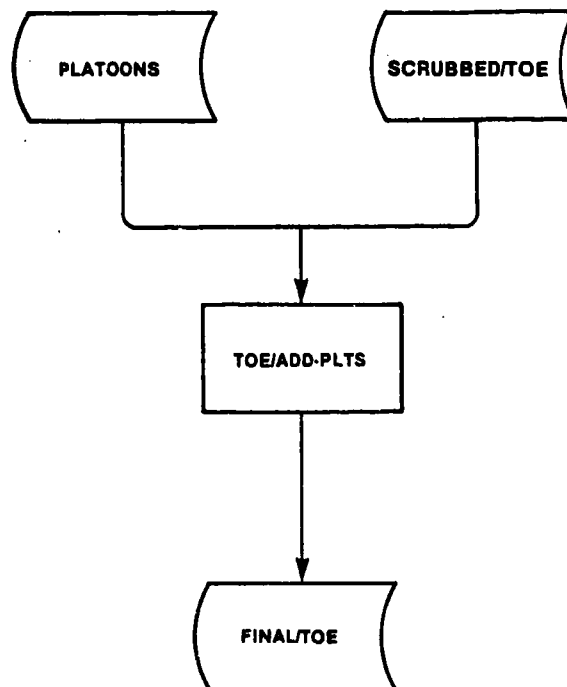


Figure III.10.1

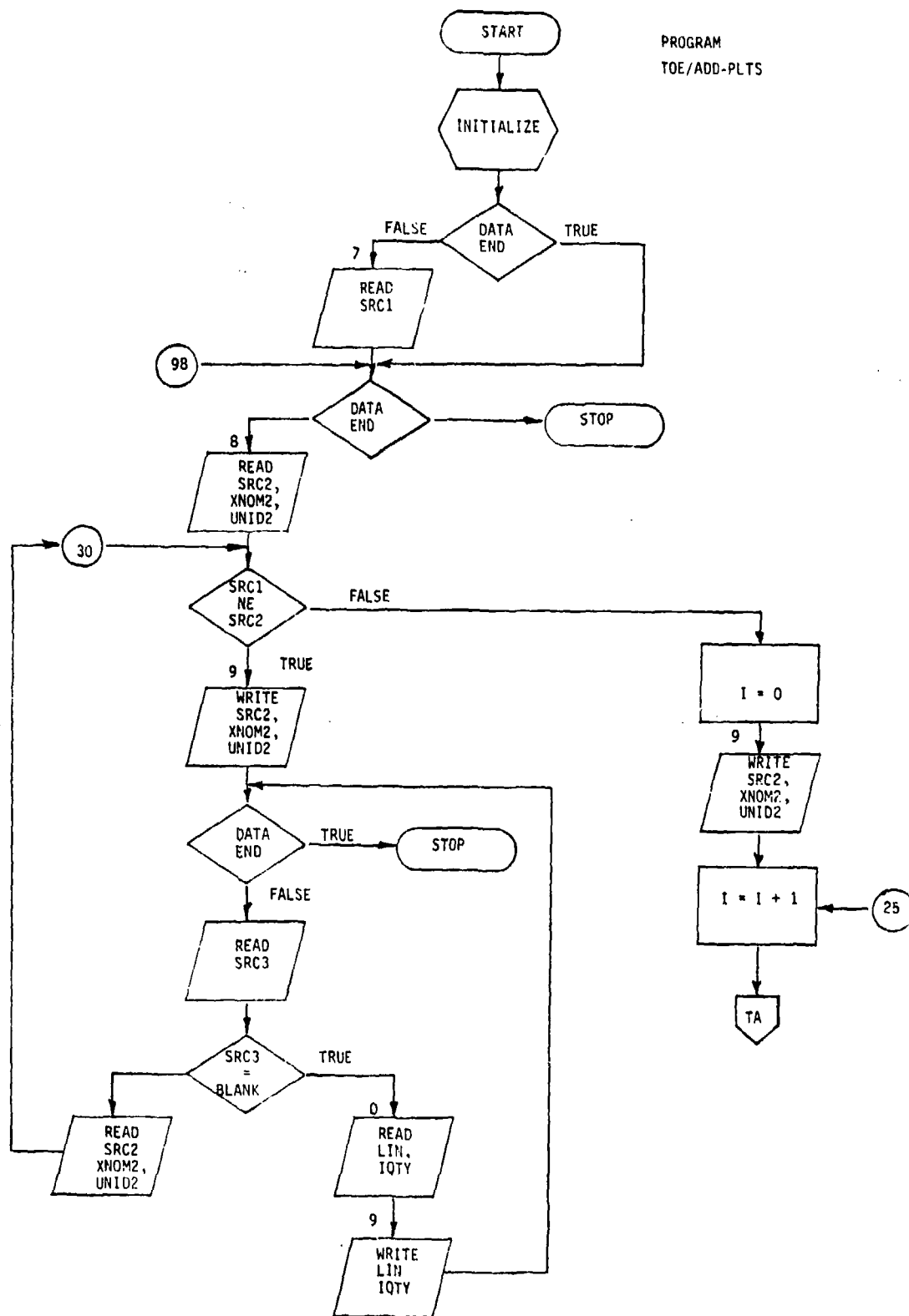


Figure III.10.2

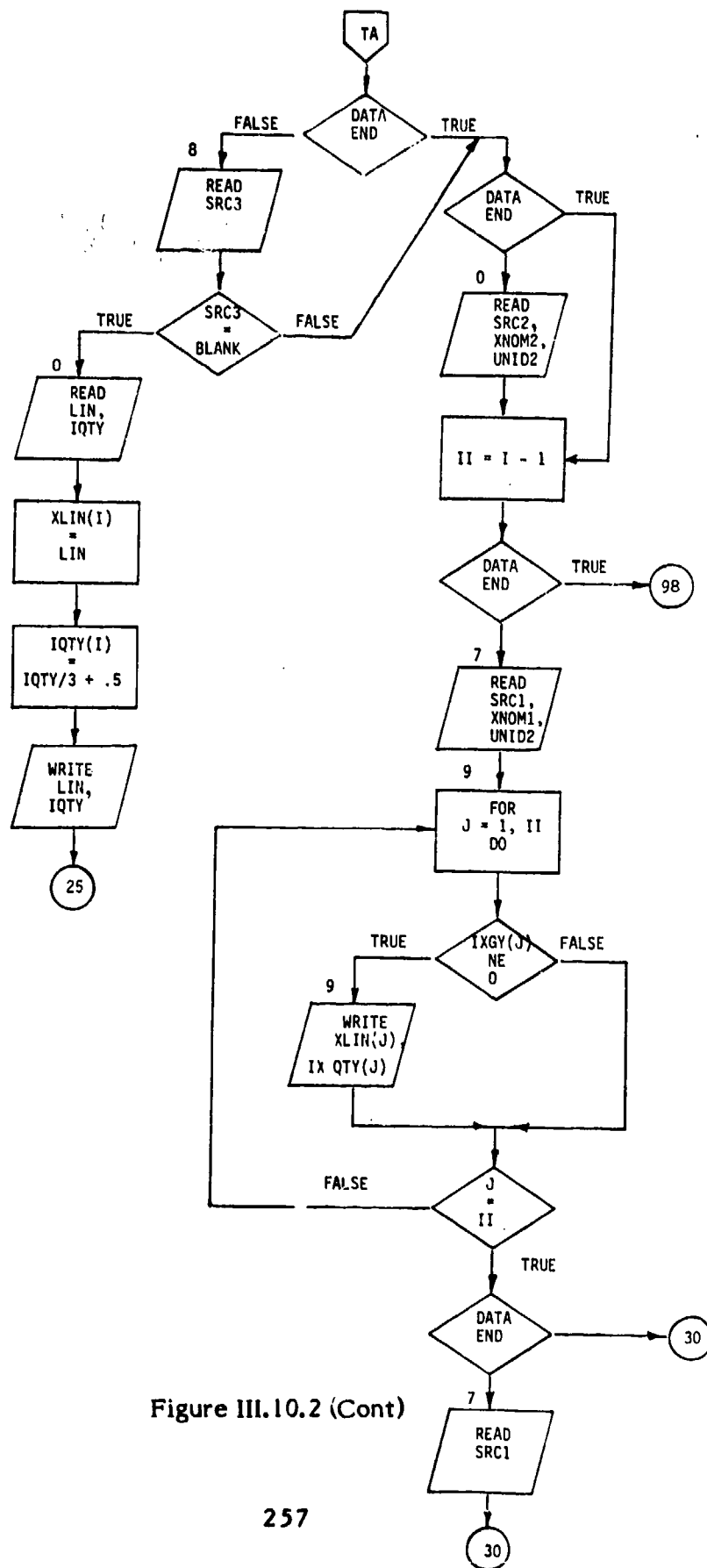


Figure III.10.2 (Cont)

UNCLASSIFIED\*\*\*FILE NAME:82XQT ELEMENT NAME:TOE/ADD-PLTS\*\*\*UNCLASSIFIED

```

1:C*****
2:C*****
3:C*****THIS PROGRAM IS WRITTEN IN ASCII FORTRAN II.
4:C*****
5:C*****
6:C*****PURPOSE OF PROGRAM:
7:C*****
8:C*****THIS UTILITY PROGRAM IS DESIGNED TO CREAT A SEPARATE TOE
9:C*****LIN CODE AND QUANTITY LISTING FOR PLATOON SIZE UNITS FROM
10:C*****THE TOE DATA PROVIDED BY THE TOE UTILITIES "82XQT.TOE/1STRUN" AND
11:C*****"82XQT.SCRUB/TOE".
12:C*****A INPUT FILE WITH PARENT UNIT SRC,UNIT TITLE AND SEQUENCE
13:C*****NUMBER FOLLOWED BY THE SAME INFO FOR THE PLATOON SIZE UNIT
14:C*****MUST BE CREATED BY MANUAL EDIT. THE OTHER INPUT FILE WILL BE THE
15:C*****OUTPUT FILE OF THE UTILITY "82XQT.SCRUB/TOL".
16:C*****THE OUTPUT FILE OF THIS UTILITY WILL BE THE FINAL TOE FILE
17:C*****USED TO CREAT THE WARFRAM MATRIX QUANTITIES BY UNIT AND
18:C*****VULNERABILITY CATEGORY.
19:C*****THE PARENT UNIT(COMPANY) AND THE RESULTING SUBUNIT(PLATOON)
20:C*****CREATED BY THIS UTILITY MUST BE FURTHER EXAMINED AND CHANGES
21:C*****MADE IN CERTAIN LIN CODE AUTHORIZED QUANTITIES WHERE THE
22:C*****TYPE AND QUANTITY OF THESE ITEMS DOES NOT MATCH WITH THE REAL
23:C*****WORLD.
24:C*****
25:C*****
26:C*****VARIABLE DICTIONARY
27:C*****VARIABLE NAME      DEFINITION
28:C*****SRC1              SRC FROM INPUT FILE SHOWING
29:C*****                  TITLE OF UNITS MISSING TOE DATA ON
30:C*****                  PLATOON SIZE UNITS.
31:C*****
32:C*****SRC2              SRC FROM TOE MASTER FILE
33:C*****
34:C*****XNUMM1            TITLE OF UNIT MATCHING SRC1
35:C*****
36:C*****XNUMM2            TITLE OF UNIT MATCHING SRC2
37:C*****
38:C*****SRC3              USED TO CHECK FOR AN SRC NUMBER
39:C*****
40:C*****LIN              READ FROM TOE MASTER FILE
41:C*****
42:C*****XLIN(I)           USED TO STORE LIN CODE FOR OUTPUT
43:C*****                  TO FINAL TOE STUDY FILE
44:C*****
45:C*****IXQTY(I)          COMPUTED PLATOON LIN CODE QUANTITY
46:C*****
47:C*****UNID1             SEQUENCE NUMBER OF UNIT IN FILE UNIT 7
48:C*****
49:C*****UNID2             SEQUENCE NUMBER OF UNIT IN FILE UNIT 8
50:C*****
51:C*****
52:      DIMENSION IXQTY(3000)
53:      CHARACTER*4 UNID1
54:      CHARACTER*4 UNID2
55:      CHARACTER*9 SRC1
56:      CHARACTER*9 SRC2
57:      CHARACTER*31 XNUM1

```

Figure III.10.3

UNCLASSIFIED\*\*\*FILE NAME:82XQT ELEMENT NAME:TOE/ADD-PLTS\*\*\*UNCLASSIFIED

```

58: CHARACTER*31 XNOM2
59: CHARACTER*5 SRC3
60: CHARACTER*6 LIN
61: CHARACTER*6 XLIN(2000)
62: READ(7,100,END=98)SRC1
63:100 FORMAT(1X,A9)
64:98 READ(8,200,END=99)SRC2,XNOM2,UNID2
65:200 FORMAT(A9,5X,A31,A4)
66:30 IF(SRC1.NE.SRC2)
67: *THEN
68:35 WRITE(9,200)SRC2,XNOM2,UNID2
69:15 READ(8,201,END=99)SRC3
70:201 FORMAT(A5)
71: IF(SRC3.EQ.' ')
72: * THEN
73: READ(10,202)LIN,IQTY
74: WRITE(9,202)LIN,IQTY
75:202 FORMAT(5X,A6,2X,I4)
76: GO TO 15
77:10 READ(10,203)SRC2,XNOM2,UNID2
78: GO TO 30
79:20 I=0
80: WRITE(9,200)SRC2,XNOM2,UNID2
81:25 I=I+1
82: READ(10,201,END=40)SRC3
83: IF(SRC3.EQ.' ')
84: * THEN
85: READ(10,202)LIN,IQTY
86: XLIN(I)=LIN
87: IXQTY(I)=IQTY/3+.5
88: WRITE(9,202)LIN,IQTY
89: GO TO 25
90:40 READ(10,200,END=45)SRC2,XNOM2,UNID2
91:45 II=I-1
92: READ(7,101,END=98)SRC1,XNOM1,UNID1
93:101 FORMAT(1X,A9,2X,A25,A4)
94: WRITE(9,102)SRC1,XNOM1,UNID1
95:102 FORMAT(A9,5X,A31,A4)
96: DO 100 J=1,II
97: IF(IXQTY(J).NE.0)
98: * THEN
99: WRITE(9,202)XLIN(J),IXQTY(J)
100: ELSE
101: GO TO 50
102: ENDOF
103:50 CONTINUE
104: ELSE
105: GO TO 40
106: ENDOF
107: ELSE
108: GO TO 10
109: ENDOF
110: ELSE
111: GO TO 20
112: ENDOF
113: READ(7,100,END=30)SRC1
114: GO TO 30
115:99 STOP
116: END

```

Figure III.10.3 (Cont)

UNCLASSIFIED\*\*\*EXAMPLE OF OUTPUT OF UTILITY SCRUB/TOF\*\*\*UNCLASSIFIED

1:03097H700	N90 DEF CO	CH01
2:	A32444	9
3:	E00533	12
4:	E45820	1
5:	E70064	1
6:	F81880	9
7:	J43918	11
8:	J46110	1
9:	L44595	10
10:	M75714	13
11:	N96741	1
12:	P43177	2
13:	P95592	9
14:	G19339	18
15:	Q20935	38
16:	Q21483	19
17:	Q53001	4
18:	Q54174	4
19:	Q56783	6
20:	Q78292	1
21:	R94977	110
22:	U01305	2
23:	V15018	9
24:	V31211	6
25:	W32593	1
26:	W95400	13
27:	W95811	11
28:	X39447	1
29:	X40146	11
30:	X40968	9
31:	X58367	1
32:	X60833	13
33:	X63299	1
34:05145H710	CBT ENG BN	END2
35:	A32444	10
36:	983582	2
37:	B83856	18
38:	984404	27
39:	C20414	6
40:	C25757	2
41:	C85494	3
42:	C86213	1
43:	D11538	3
44:	D12037	40
45:	E00533	24
46:	E45820	8
47:	E56578	8
48:	E63242	1
49:	E70064	5
50:	F70686	1
51:	F73626	2
52:	F39376	3
53:	F81880	1
54:	G02204	48
55:	G02341	50
56:	H02300	2
57:	H38787	2

Figure III.10.4

UNCLASSIFIED\*\*\*EXAMPLE OF INPUT DATA FILE PLATOONS\*\*\*UNCLASSIFIED

1:	07047H010	MECH INF CO	ME25
2:	07047H9X9	MECH INF PLT	ME26
3:	17037H010	TANK 105MM CO	AR51
4:	17037H9X9	TANK 105MM PLT	AR52
5:	17307H700	ARMD CAV TRP	CA58
6:	17307H9X9	ARMD CAV PLT	CA59
7:	44267H500	HAWK BRTY	AD74
8:	44267H9X9	HAWK PLT	AD75
9:	44327H000	ADA BRTY VULCAN	AD78
10:	44327H9X9	ADA PLT VULCAN	AD79
11:	44323H000	ADA BTRY CHAPARRAL	AD80
12:	44328H9X9	ADA PLT CHAPARRAL	AD81

Figure III.10.5

UNCLASSIFIED \*\*\*EXAMPLE OF OUTPUT OF UTILITY YOL/ADD-PLTS\*\*\*UNCLASSIFIED

	ABC DEF CO	CH01
1:030070700		
2:	A32444	4
3:	E00533	12
4:	E45820	1
5:	E70064	1
6:	F81660	9
7:	J43918	11
8:	J46110	1
9:	L44595	10
10:	M75714	13
11:	M96741	1
12:	M43177	2
13:	M95592	9
14:	N19339	18
15:	N20935	38
16:	N21483	19
17:	N53001	4
18:	N54174	4
19:	N56783	6
20:	N74482	1
21:	N94977	110
22:	O01305	2
23:	O15018	9
24:	O31211	6
25:	O32593	1
26:	O95400	13
27:	O95511	11
28:	A39447	1
29:	A40146	11
30:	A40768	9
31:	A58467	1
32:	A60833	13
33:	A63299	1
34:051450710	CH1 F00 HK	EN02
35:	A32444	10
36:	B83582	2
37:	B83856	18
38:	B44404	27
39:	C20414	6
40:	C25757	2
41:	C85494	3
42:	C86213	1
43:	O11538	3
44:	O14067	40
45:	E00533	24
46:	E45820	8
47:	E56578	8
48:	E69242	1
49:	E70064	5
50:	E70886	1
51:	E73826	2
52:	F37378	3
53:	F81660	1
54:	G02204	48
55:	G02341	50
56:	G02300	2
57:	H38757	2

Figure III.10.6

## CHAPTER 11

### UTILITY - RAM/MATRIX OR WIMP/MATRIX

**11.1 DESCRIPTION OF PROCESSING:** This program will be titled WIMP/MATRIX when the use of the WIMP is involved in the methodology. The program sums the equipment items by vulnerability category. Otherwise the functions are logic tested read and write statements.

**11.1.1 PURPOSE/FUNCTIONS:** The purpose of this utility is to process and merge data from three files and develop the RAM/MATRIX. The RAM/MATRIX is a two dimensional array. The Y axis of the array is the units in the study identified by their two character unit codes and a sequence number. The X axis of the matrix consists of 22 columns; one for each of the 22 vulnerability categories into which an item of equipment will be classified. These categories are used to group together items of equipment which have common susceptibilities to indirect fire.

The interior of the matrix accumulates for each unit in the study the number of items of equipment the unit is authorized in each of the vulnerability categories.

This RAM/MATRIX data file of unit type and item vulnerability category is used as input to other modules such as the Target Acquisition Model (TAM) output files, the TOTAL/CATEGORY utility and the TOTAL/UNITS utility.

**11.1.2 PROGRAM INPUT/OUTPUT STRUCTURE:** The overall structure of the utility is pictured in Figure III.11.1. The logic followed by the utility is pictured in FIGURE III.11.2. The source code for the utility is listed in FIGURE III.11.3.

**11.1.2.A INPUT DATA AND DATA BASE:** This utility uses three data files as its input. One file is the ARRAYED/UNITS file which was prepared manually by the study analyst. The second file is the ITMID/FINAL file which was produced by the ITMID/REC-A utility. The final input file is the FINAL/TOE file which was produced by the TOE/ADD-PLTS utility.

From these files the utility extracts data on:

- o The type units participating in the study Stylized Array (ARRAYED/-UNITS)
- o Items of equipment being used in the study and their artillery vulnerability category (ITMID/FINAL).
- o The authorized allocations of items of equipment to units (FINAL/TOE).

Figures III.11.4 and III.11.5 depict the file layout and example data for the ARRAYED/UNITS plan; Figures 3.9.5 and 3.9.7 present the file layout and example data for the ITMID/FINAL file; and Figures 3.9.8 and 3.9.9 present the file layout and example data for the FINAL/TOE file.

**11.1.2.B OUTPUT DATA AND DATA FILES:** - The RAM/MATRIX file is the only output from the RAM/MATRIX utility. This file has one record type consisting of the unit identification code and the 22 occurrences of the artillery vulnerability categories. As the utility executes it examines for each unit type each individual item of equipment which it has been authorized and determines the equipment's artillery vulnerability category and the quantity of this equipment this unit has been authorized and accumulates in the vulnerability category for this unit the quantity of items authorized. It should be noted that the emphasis in this process is on unit type and number of equipment units in specific vulnerability categories, regardless of the specific type of equipment.

Figures III.11.10 and III.11.11 present the file layout and examples of the data found in the file.

### **11.12.C DATA ELEMENT DICTIONARY:**

The following section will identify and define all variables used in this utility.

<u>NAME</u>	<u>DEFINITION</u>
LIN1	A 6 character alphanumeric variable which identifies an item of equipment within the process. The utility has established this variable as an array. Data used to fill the array will be provided by the ITMID/FINAL file.
LIN2	Same as LIN1 except the variable is not used as an array and its source of data is the FINAL/TOE file.
IVUL	A 2 character integer field which denotes the artillery vulnerability category in which the item of equipment has been placed. There are 22 classes of vulnerability. This data is supplied by the ITMID/FINAL file. This variable is dimensioned as array by the utility.
CODE	This is a 2 character alphanumeric variable which is used to hold the first 2 characters of the 4 character UNIT ID code. This variable is dimensioned as an array. It is provided by the ARRAYED/UNITS file.
DUMMY	This is a 72 character field into which is ready data not needed by the utility from the ITMID/FINAL file.
ITEST	This is a 2 character integer variable. It is used when reading the FINAL/TOE file in order to determine whether the current record is a header record containing unit identification data, or a detail record which contains data on items of equipment and quantity authorizations.

NCAT	This is an integer variable which is used to accumulate for each of the 22 artillery vulnerability categories, the total quantity of units of equipment being played in the study and assigned this vulnerability category. This variable is calculated in the utility and written out to the RAM/MATRIX file. This variable is dimensioned as an array of 22 members.
I	This is an integer variable which is used to count records from the ITMID/FINAL file as they are read.
NLIN	An integer value which will hold the total number of records in the ITMID/FINAL file. NLIN is equal to I-1. It is used as the upper limit in a following DO LOOP.
MM	An integer variable which is used as a subscript in the IVUL array. It is set to the proper vulnerability category of the item of equipment being processed.
M, II, N, NN	Various integer variables used as subscripts in the utility.

**11.2 OPERATING ENVIRONMENT:** This program is implemented on the EXECUTIVE-8 operating system.

**11.2.1 SUPPORT SOFTWARE:** - This utility requires the ASCII FORTRAN compiler and the UNIVAC 1100/82 system facilities.

**11.2.2 I/O DEVICES:** - This utility uses as input files which are resident on disk and in turn produce as output files that also will reside on disk. Refer to Volume I for the program runstream.

### **11.3 MAINTENANCE PROCEDURES:**

**11.3.1 PROGRAMMING CONVENTIONS:** This utility is written in ASCII FORTRAN and takes advantage of the IF-THEN-ELSE structuring available. The logic structure is somewhat disguised, however, because of the lack of indentation.

**11.3.2 INTERNAL ERROR ROUTINES:** - There are no explicit error handling routines written into the utility. As a result only system detected errors will be identified and the use of available system documentation will be required to correct the error condition.

# RAM/MATRIX STRUCTURE

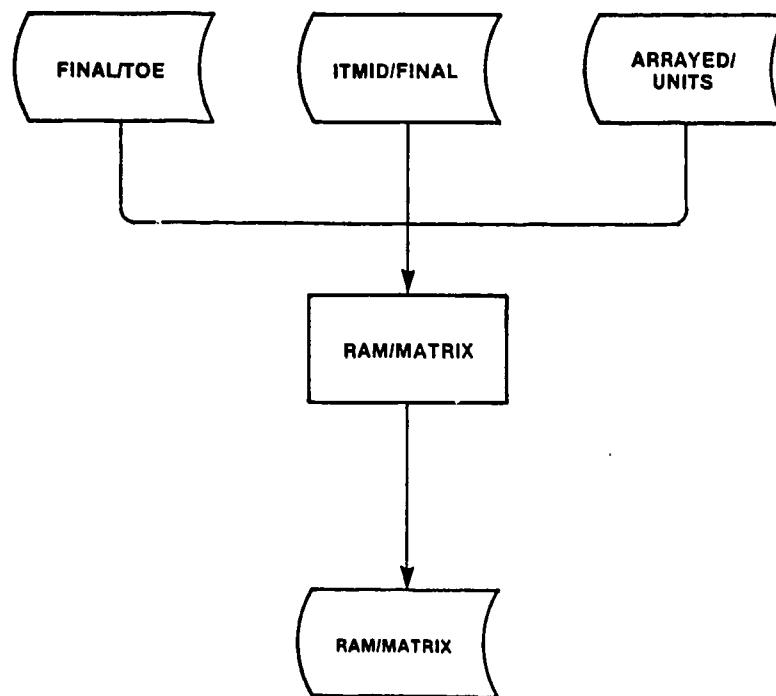


Figure III.11.1

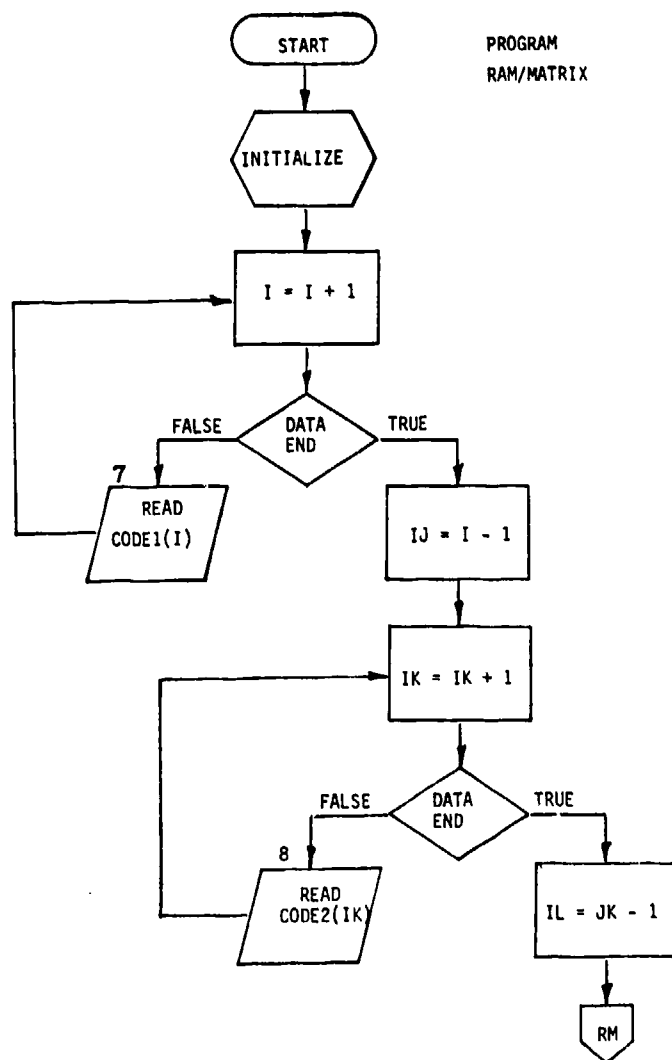


Figure III.11.2

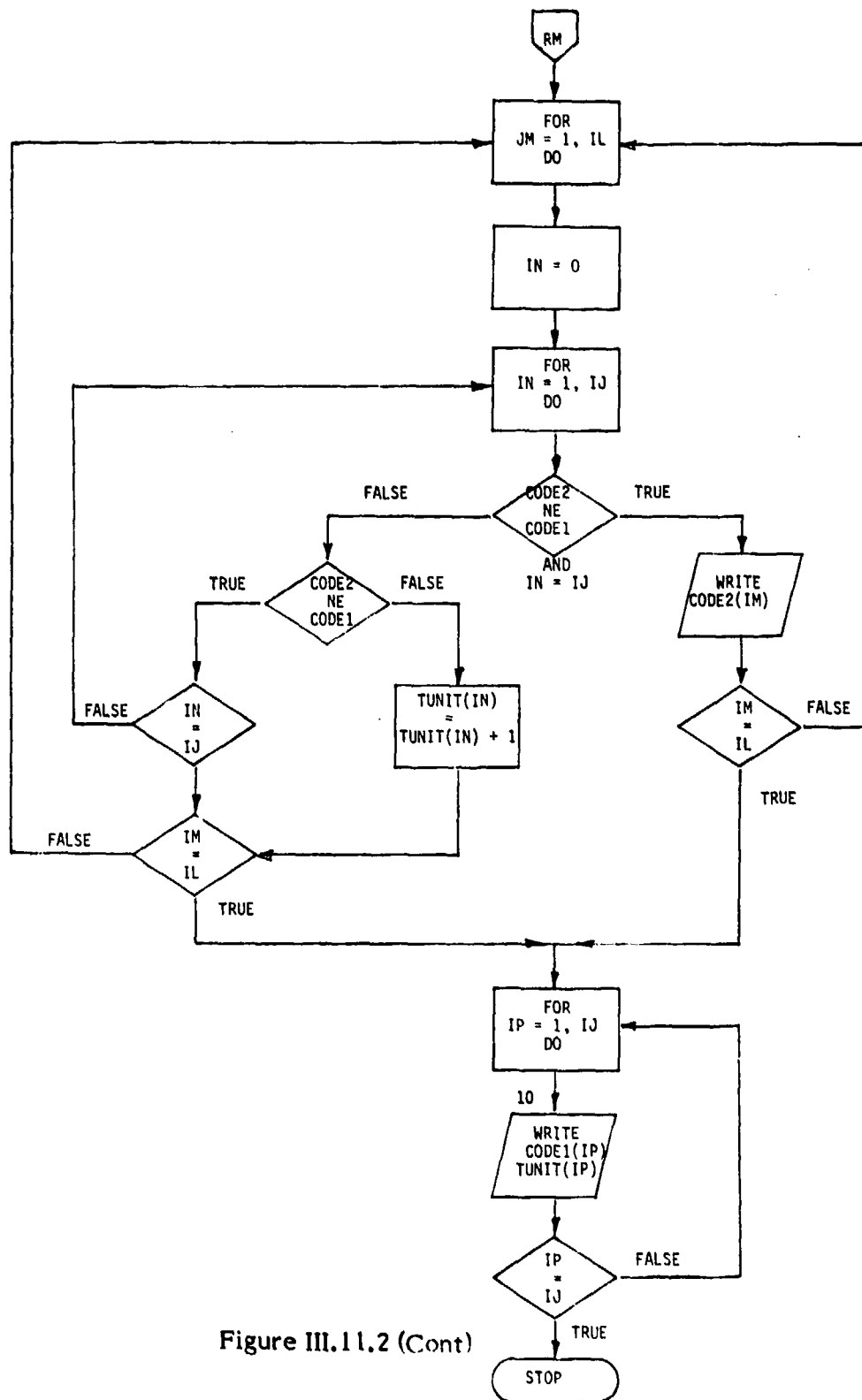


Figure III.11.2 (Cont)

UNCLASSIFIED\*\*\*FILE NAME:82XGT ELEMENT NAME:RAM/MATRIX\*\*\*UNCLASSIFIED

```

1:C*****
2:C*****
3:C*****THIS PROGRAM IS WRITTEN IN ASCII FORTRAN.
4:C*****
5:C*****
6:C*****THIS UTILITY IS DESIGNED TO CREAT THE WARFRAM (UNIT BY VULNER-
7:C*****ABILITY CATEGORY) MATRIX FROM THE FINAL TOE FILE , ITMID FILE
8:C*****AND THE LIN CODE FILE. THE OUTPUT OF THIS UTILITY WILL SERVE
9:C*****AS INPUT TO THE TAM DATA FILE, PROVIDING THE AUTHORIZED QUANTITIES
10:C*****OF EQUIPMENT(LIN CODES) FOR EACH TYPE OF UNIT BY 22 ARTILLERY VUL-
11:C*****NERABILITY CATEGORIES. THE RAM WILL USE THIS DATA TO CALCULATE
12:C*****TOTAL LOSSES ATTRIBUTED TO RED ARTILLERY FIRE DURING A 24 HOUR
13:C*****PERIOD (STYLIZED DAY) FOR EACH OF THE 22 ARTILLERY VULNERABILITY
14:C*****CATEGORIES AND EACH OF THE FOUR STYLIZED COMBAT POSTURES.
15:C*****
16:C*****
17:C*****VARIABLE DICTIONARY
18:C*****VARIABLE NAME      DEFINITION
19:C      LIN1(I)            LIN CODE FROM ITMID FILE
20:C
21:C      IVULN(I)           VULNERABILITY CODE FOR LIN1(I)
22:C
23:C      DUMMY(I)           USED TO READ PAST UNUSED DATA
24:C
25:C      ITEST              USED TO CHECK FOR UNIT SRC AND TITLE LINE
26:C
27:C      LIN2               LIN CODE FROM THE TOE DATA FILE
28:C
29:C      IQTY               QUANTITY FOR LIN CODE FROM TOE FILE
30:C
31:C      MM                 USED TO SET NCAT(MM)TO THE CORRECT VULNER-
32:C                      ABILITY CATEGORY(1-22) FOR WARFRAM
33:C
34:C      NCAT(MM)           VULNERABILITY CATEGORY EQUIPMENT TOTALS
35:C
36:C      NLIN               EQUALS TOTAL NUMBER OF LIN CODES READ IN
37:C                      FROM ITMID FILE
38:C*****
39:C*****
40:      DIMENSION IVULN(2000),NCAT(22)
41:      CHARACTER*6 DUMMY(12)
42:      CHARACTER*6 LIN1(2000)
43:      CHARACTER*6 LIN2
44:      CHARACTER*2 CODE(200)
45:10      III=III+1
46:      READ(7,700,END=20)CODE(III)
47:700      FORMAT(37X,A2)
48:      GO TO 10
49:20      I=I+1
50:      READ(8,200,END=40)LIN1(I),IVULN(I)
51:200      FORMAT(1X,A6,34X,I2)
52:      DO 30 K=1,8
53:      READ(8,201,END=40)(DUMMY(L),L=1,12)
54:201      FORMAT(12A6)
55:30      CONTINUE
56:      GO TO 20
57:40      NLIN=I-1

```

Figure III.11.3

UNCLASSIFIED\*\*\*FILE NAME:82XGT ELEMENT NAME:RAM/MATRIX\*\*\*UNCLASSIFIED

```
58:      READ(9,203,END=99)ITEST
59:203   FORMAT(47X,I2)
60:45    READ(9,203,END=99)ITEST
61:      IF(ITEST.EQ.0)
62:      *THEN
63:      READ(10,204,END=99)LIN2,IQTY
64:204   FORMAT(5X,A6,2X,I4)
65:      DO 50 M=1,NLIN
66:      IF(LIN1(M).EQ.LIN2)
67:      *THEN
68:      MM=IVULN(M)
69:      NCAT(MM)=NCAT(MM)+IQTY
70:      ENDIF
71:50     CONTINUE
72:      ELSE
73:      II=II+1
74:      WRITE (10,300)CODE(II),I I,(NCAT(N),N=1,22)
75:300    FORMAT(1X,A2,I2,22I4)
76:      DO 60 NN=1,22
77:      NCAT(NN)=0
78:60     CONTINUE
79:      ENDIF
80:      GO TO 45
81:99    II=II+1
82:      WRITE (10,300)CODE(II),I I,(NCAT(N),N=1,22)
83:      STOP
84:      END
```

Figure III.11.3 (Cont)

UNCL 0511100000EXAMPLE OF OUTPUT OF UTILITY YOL/ADD-PLTS00UNCLASSIFIED

11030270700	HBC DEF CO	CH01
21	432444	4
31	000533	12
41	445820	1
51	470064	1
61	481360	9
71	043918	11
81	046110	1
91	444595	10
101	075714	13
111	096741	1
121	043177	2
131	495392	9
141	419339	18
151	420935	38
161	421483	19
171	453001	4
181	454174	4
191	456783	6
201	474262	1
211	094977	110
221	001305	2
231	015018	9
241	031211	6
251	032593	1
261	095400	13
271	095411	11
281	439447	1
291	440146	11
301	440968	9
311	458367	1
321	460833	13
331	463299	1
34:051450710	CH1 FNG BR	END2
351	032444	10
361	043582	2
371	063856	18
381	084404	27
391	020414	6
401	025757	2
411	035494	3
421	066713	1
431	011530	3
441	014087	40
451	000533	24
461	445820	8
471	456578	8
481	469242	1
491	470064	5
501	470066	1
511	473826	2
521	437378	3
531	481080	1
541	482204	48
551	482341	50
561	482300	2
571	483767	2

Figure III.11.4

UNCLASSIFIED\*\*\*EXAMPLE OF ARRAYED/UNITS DATA FILE\*\*\*UNCLASSIFIED

1:	03087H700	NBC DEF CO	CM01
2:	05145H710	CBT ENG BN	EN02
3:	05146H700	HHC ENG BN	EN03
4:	05147H000	ENG CO	EN04
5:	05148H710	DRG CO	EN05
6:	06302H000	HMB DIVARTY	FA06
7:	06365H000	155MM SP BN	FA07
8:	06366H000	HMD 155MM BN	FA08
9:	06367H000	155MM BTRY	FA09
10:	06369H000	SVC BTRY 155MM	FA10
11:	06395H110	8"/GSRS BN	FA11
12:	06396H110	HWP 8"/GSRS BN	FA12
13:	06397H000	8"BTRY	FA13
14:	06398H100	GSRS BTRY	FA14
15:	06399H000	SVC BTRY 8"/GSRS	FA15
16:	06445H100	8" SF BN(CORPS)	FA16
17:	06446H100	HWP 8" BN(CORPS)	FA17
18:	06447H100	8" BTRY(CORPS)	FA18
19:	06449H100	SVC BTRY 8"(CORPS)	FA19
20:	06515H000	GSRS BN(CORPS)	FA20
21:	06516H000	HWP GSRS BN(CORPS)	FA21
22:	06517H000	GSRS BTRY(CORPS)	FA22
23:	07045H020	MECH INF BN	ME23
24:	07046H010	HHC MECH INF BN	ME24
25:	07047H010	MECH INF CO	ME25
26:	07047H9X9	MECH INF PLT	ME26
27:	07049H020	CS CO INF	ME27
28:	08035H000	MED BN	MD28
29:	08036H000	HHC MED BN	MD29
30:	08037H000	MED CO	MD30
31:	08123H000	CBT SUPT HOSRITAL (CORPS)	MD31
32:	08127H410	MED AMP CO(CORPS)	MD32
33:	08137H200	MED AIR AMP CO(CORPS)	MD33
34:	09038H300	ORO CO CONV AMMO(CORPS)	OR34
35:	09047H400	SPEC AMMO DS CO(CORPS)	OR35
36:	09048H900	SPEC AMMO DS/OS CO(CORPS)	OR36
37:	09268H900	MNT BTRY HAWK(CORPS)	AD37
38:	09557H510	MISSILE SUPT CO	AD38
39:	10007H000	SBS CO	QM39
40:	10207H300	PETRL PL&TML OP CO(CORPS)	QM40
41:	11035H000	SIGNAL BN	SC41
42:	11036H000	HHC SIG BN	SC42
43:	11037H000	CMD OPS CO	SC43
44:	11038H000	FWD COMM CO	SC44
45:	11039H000	SIG SUPT OP CO	SC45
46:	12017H610	AG CO	AC46
47:	14037H610	FIN CO	FC47
48:	17004H000	HHC ARMD DIV	AR48
49:	17035H010	TANK 105MM BN	AR49
50:	17036H000	HHC TANK BN	AR50
51:	17037H010	TANK 105MM CO	AR51
52:	17037H9X9	TANK 105MM PLT	AR52
53:	17039H000	CS CO TANK	AR53
54:	17042H000	HHC ARMD BDE	CA54
55:	17105H020	ARMD CAV SQDN	CA55
56:	17106H000	HMT CAV SQDN	CA56
57:	17108H000	ATR CAV TRP	CA57

Figure III.11.5

UNCLASSIFIED\*\*\*EXAMPLE OF THE OUTPUT OF UTILITY ITMD/REC-A\*\*\*UNCLASSIFIED

1:	AD3193	AK	VEH	M210	GM	EG	P1A			U 522	30	0	1	2	1
2:	54	64	44	58	78	88	88								
3:	.00	.00	.00	.80	.20										
4:	.00	.00	.00	.90	.20										
5:	.00	.00	.00	.80	.20										
6:	.00	.00	.00	.80	.20										
7:	.00	.00	.00	.80	.20										
8:	.00	.00	.00	.80	.20										
9:	.00	.00	.00	.75	.25										
10:	A14752	ADAP	TEST	CAMERA	LM178					U1636	30	0	1	2	2
11:	10	16	16	17	17					17	17				
12:	.00	.00	.20	.80	.00										
13:	.00	.00	.20	.80	.00										
14:	.00	.00	.20	.80	.00										
15:	.00	.00	.15	.85	.00										
16:	.00	.00	.15	.85	.00										
17:	.00	.00	.35	.65	.00										
18:	.00	.00	.35	.65	.00										
19:	A23490	AIMING	CIRCLE	M2	W/E					U1636	30	0	1	2	3
20:	8015	6699	9820	8923	8823	8923	8978								
21:	.25	.25	.50	.00	.00										
22:	.25	.25	.50	.00	.00										
23:	.25	.25	.50	.00	.00										
24:	.25	.25	.50	.00	.00										
25:	.25	.25	.50	.00	.00										
26:	.25	.25	.50	.00	.00										
27:	.25	.25	.50	.00	.00										
28:	A23770	AIR	COND	FL/WNDW	6000R					U1833	30	0	1	2	4
29:	0	0	0	0	0					0	0				
30:	.00	.00	.00	.00	.00										
31:	.00	.00	.00	.00	.00										
32:	.00	.00	.00	.00	.00										
33:	.00	.00	.00	.00	.00										
34:	.00	.00	.00	.00	.20										
35:	.00	.00	.00	.00	.00										
36:	.00	.00	.00	.00	.00										
37:	A23928	AIR	COND	F/WA	9000	BTU				U1833	30	0	1	2	5
38:	889	993	998	998	998	998				998	998				
39:	.00	.00	.25	.25	.50										
40:	.00	.00	.25	.25	.50										
41:	.00	.00	.25	.25	.50										
42:	.00	.00	.25	.25	.50										
43:	.00	.00	.25	.25	.50										
44:	.00	.00	.25	.25	.50										
45:	.00	.00	.25	.25	.50										
46:	A24044	AIR	COND	18000	BTU					U1833	30	0	1	2	6
47:	53	53	53	55	55					55	55				
48:	.00	.00	.00	.50	.50										
49:	.00	.00	.00	.50	.50										
50:	.00	.00	.00	.50	.50										
51:	.00	.00	.00	.50	.50										
52:	.00	.00	.00	.30	.70										
53:	.00	.00	.00	.30	.70										
54:	.00	.00	.00	.30	.70										
55:	A24318	AIR	COND	18000	BTU					U1833	30	0	1	2	7
56:	15	25	42	57	63					83	97				
57:	.00	.00	.25	.75	.00										

Figure III.11.6

## UNCLASSIFIED\*\*\*EXAMPLE OF THE OUTPUT DATA OF UTILITY RAM/MATRIX\*\*\*UNCLASSIFIED

1: CH01	0	0	0	23	22	0	74	0	0	0	0	0	121	13	87	24	73	1	0	0	9
2: EN02	0	0	4	55	88	0	86	0	20	10	0	0	41142	79	315	271	101	147	26	0	15
3: EN03	0	0	0	24	32	0	24	0	7	2	0	0	4 202	9	55	71	43	11	7	0	15
4: EN 4	0	0	0	8	9	0	12	0	2	2	0	0	0 194	15	51	36	9	30	4	0	C
5: EN05	0	0	4	11	20	0	14	0	5	0	0	0	0 160	6	56	56	72	16	3	0	0
6: FA06	0	0	0	35	14	0	17	0	2	0	0	0	0 216	10	53	131	17	25	1	0	0
7: FA07	0	18	0	43	39	0	24	0	5	0	2	94	2 F45	74	136	356	78	118	3	0	0
8: FA08	0	0	0	29	3	0	13	0	1	0	0	0	U 240	17	51	234	43	73	2	0	0
9: FA09	U	6	0	3	6	0	2	0	1	0	0	15	0 108	17	22	34	8	13	0	0	0
10: FA10	0	0	0	5	18	0	5	0	1	0	2	9	2 91	6	19	20	11	6	1	0	0
11: FA11	0	19	0	59	54	0	37	0	21	1	2	48	0 767	82	170	416	115	106	7	0	0
12: FA12	0	1	0	34	4	0	15	0	1	0	0	0	0 234	70	61	224	53	58	1	0	0
13: FA13	0	4	0	4	8	0	3	0	1	0	0	12	0 105	14	21	37	8	10	1	0	0
14: FA14	0	6	0	7	3	0	4	0	1	0	0	0	0 94	11	26	58	25	11	1	0	0
15: FA15	0	0	0	6	23	0	9	0	16	1	2	12	0 134	9	70	23	13	7	2	0	0
16: FA16	0	12	0	42	42	0	28	0	8	1	2	42	0 544	57	123	210	93	64	1	0	0
17: FA17	0	0	0	27	3	0	16	0	1	0	0	0	0 145	11	41	93	19	21	0	0	0
18: FA18	0	4	0	3	7	0	2	0	1	0	0	11	C 104	13	22	32	8	12	0	0	0
19: FA19	0	0	0	6	18	0	6	0	1	1	2	9	0 87	7	16	21	10	7	1	0	0
20: FA20	0	30	0	74	74	0	40	0	58	0	2	0	0 542	19	82	321	101	23	1	0	0
21: FA21	0	0	0	24	11	0	19	0	1	0	2	0	0 125	4	19	75	26	14	1	0	0
22: FA22	0	10	0	18	21	0	7	0	19	0	0	0	0 139	5	21	82	25	3	0	0	0
23: ME23	0	13	0	34	31	0	46	0	5	0	6	0	2 976	142	459	452	43	87	3	0	0
24: ME24	0	0	0	14	23	0	19	0	8	0	2	0	2 184	21	43	74	29	15	3	0	0
25: ME25	0	3	0	3	2	0	5	0	0	0	1	0	0 208	35	93	95	3	12	0	0	0
26: ME26	0	1	0	1	0	0	1	0	0	0	0	0	0 68	11	30	28	0	2	0	0	0
27: ME27	0	4	0	11	2	0	12	0	0	0	1	0	0 168	16	137	93	5	16	0	0	0
28: MD28	0	0	0	69	35	0	58	0	8	0	0	0	0 400	0	58	80	50	16	4	0	0
29: MD29	0	0	0	27	14	0	28	0	2	0	0	0	0 154	0	22	41	23	4	1	0	0
30: MD30	0	0	0	14	7	0	10	0	2	0	0	0	0 82	0	12	13	9	4	1	0	0
31: MD31	0	0	0	6	21	0	30	0	2	0	0	0	0 126	0	16	34	16	2	62	0	1
32: MD32	0	0	0	45	2	0	6	0	1	0	0	0	0 102	0	15	22	13	3	0	0	0
33: MD33	0	0	0	9	9	0	17	0	2	0	0	0	0 190	0	72	241	34	57	2	0	0
34: CR34	0	0	0	8	40	0	13	0	6	0	0	0	0 261	4	21	59	25	5	4	0	0
35: CR35	0	0	0	11	43	0	27	0	3	0	0	0	0 199	12	17	59	16	10	0	0	0
36: CR36	0	0	0	2	35	0	20	0	13	0	0	0	1 171	9	47	44	23	3	8	0	0
37: AD37	0	0	0	11	23	0	10	0	3	0	0	0	0 129	4	14	38	13	1	7	0	0
38: AD38	0	0	0	23	17	0	16	0	6	0	0	0	0 119	3	12	44	70	2	11	0	0
39: GM39	0	0	0	9	14	0	12	0	6	0	0	0	0 129	4	12	18	10	4	0	6	0
40: GM40	0	0	0	11	24	0	13	0	5	0	0	0	0 157	3	23	89	28	5	2	2	0
41: SC41	0	0	0	162	45	0	59	0	4	0	0	0	0 709	10	63	952	199	698	9	0	0
42: SC42	0	0	0	13	18	0	19	0	1	0	0	0	0 104	1	17	109	24	3	9	0	0
43: SC43	0	0	0	52	10	0	17	0	1	0	0	0	0 234	4	16	352	60	224	0	0	0
44: SC44	0	0	0	47	5	0	8	0	1	0	0	0	0 182	4	15	217	45	132	0	0	0
45: SC45	0	0	0	50	12	0	15	0	1	0	0	0	0 189	1	15	274	40	319	0	0	0
46: AG46	0	0	0	7	4	0	8	0	2	0	0	0	0 259	4	7	17	7	2	1	0	0
47: FC47	0	0	0	2	2	0	3	0	0	0	0	0	0 94	3	7	13	6	1	0	0	0
48: AR48	0	0	0	30	5	0	27	0	1	0	0	0	0 131	4	29	60	29	1	2	0	0
49: AR49	0	4	2	44	37	0	47	0	5	5	2	0	0 634	24	249	286	48	69	3	0	0
50: AR50	0	0	0	20	29	0	25	0	5	2	1	0	0 193	2	45	101	32	16	3	0	0
51: AR51	0	0	0	5	2	0	4	0	0	1	0	0	0 128	1	40	39	4	9	0	0	0
52: AR52	0	0	0	1	0	0	0	0	0	0	0	0	0 42	0	10	4	0	2	0	0	0
53: AR53	0	4	2	9	2	0	10	0	0	0	1	0	0 107	19	94	68	4	26	0	0	0
54: CA54	0	0	0	19	4	0	14	0	1	0	1	0	0 107	6	19	67	26	3	2	0	0
55: CA55	0	9	0	44	46	0	43	7	6	3	2	2	31079	109	540	506	70	183	6	2	0
56: CA56	0	0	0	26	74	0	19	J	5	0	2	0	3 208	39	47	89	27	23	3	0	0
57: CA57	0	0	0	6	16	0	15	0	1	0	0	2	0 225	13	103	198	19	64	3	2	0

Figure III.11.7

## CHAPTER 12

### UTILITY - TOTAL/UNITS

**12.1 DESCRIPTION OF PROCESSING:** This program performs logic tested read and write functions.

**12.1.1 PURPOSE/FUNCTIONS:** The purpose of this utility is to determine the total number of units by unit type in the RAM/MATRIX file within each of the four stylized posture arrays. These four stylized posture arrays are: Attack (AT), Defense Intense (DI), Delay (Defend) (DE) and Defense Light (Inactive) (DL). This utility must be run once for each of the four postures. The units in the posture arrays will be identified by a second input file which is created by the arrayers for the RAM analyst. Each of the four runs of this will produce two output files. One will identify units in the postured array file for which a match on a corresponding unit in the RAM/MATRIX could not be made. The second and main output from this utility is a file which contains each unit for which a match between the two files could be made and the accumulated total of the number of this type of unit that was found in the RAM/MATRIX file. Each of these four TOTAL/UNIT files will be used as input in a following utility, referred to as TOTAL/CATEGORY.

**12.1.2 PROGRAM INPUT/OUTPUT STRUCTURE:** The overall structure of the utility is seen in FIGURE III.2.1. The logic flow found in the utility is pictured in FIGURE III.12.2. The source code for the utility is listed in FIGURE III.12.3, and III.12.4 for the WIMP/TOTAL-UNITS Version.

**12.1.2.A INPUT DATA AND DATA BASE:** This utility will be run four times; once for each of the four stylized posture arrays that are used in the study. Each run of the utility will use the RAM/MATRIX file and one of the four posture array files. These files will be categorized as elements under the current study's general file. The RAM/MATRIX file will use RAM/MATRIX as its element name; the element name of the four posture files must be obtained from the RAM analyst.

The RAM/MATRIX file is produced by the RAM/MATRIX utility which was completed earlier. The purpose of the file is to identify all units by unit type appearing in the study arrays and specify for each of the 22 artillery vulnerability categories in the study, the total number of units of equipment which were grouped into each vulnerability category. The present utility concentrates on the unit type portion of the record and ignores the vulnerability category portion.

The second input file used by this utility is one of the stylized posture array files. This file is supplied by the RAM analyst who also should be contacted for the proper element name. This file is simply a listing of array units which includes the four character unit type codes which identify the unit types which can be found in this particular battle posture in the study.

Figure III.12.5 depicts the file layout and data examples for the RAM/MATRIX files. Figure III.12.6 depict the data for the stylized posture array file.

**12.1.2.B OUTPUT DATA AND DATA FILES:** - This utility produces two mass storage files as output for each of the four stylized arrays. One details those unit types which were expected to be in a particular posture as stated on the posture array input file but were not detected on the RAM/MATRIX file. This file is referred to as the MISSED/UNITS-XX file (Figure III.12.7). The XX portion of the element name must be changed to reflect the appropriate posture being analyzed in this run; i.e., AT, DI, DE, or DL. This file uses one record type consisting of one four character field, the Unit Type Code ID. The second output file of this utility is the TOTAL/UNITS-XX file. This is the major output from this utility. Once again the XX portion of the file name must be changed to reflect the posture being analyzed. This file consists of one record type. Each record is composed of 2 fields. The first field is the four character unit type code identifier. The second field in the record is a four character integer number which denotes the total number of this particular unit type that was found in the RAM/MATRIX.

For example, line 1 of Figure III.12.8, (the data example of the TOTAL/UNITS file), indicates that there were 2 units in the RAM/MATRIX file for UNIT TYPE L197 in the particular posture array.

The file is cataloged as an element under the current study's general file.

Figure III.12.7 presents the record layout and examples of the data for the MISSED/UNITS - XX file. Figures III.12.8 and III.12.9 portray the record layout and data examples for the TOTAL/UNITS-XX file from the two program sources.

#### **12.1.2.C DATA ELEMENT DICTIONARY**

The following section will identify and define all variables used in this utility.

<u>NAME</u>	<u>DEFINITION</u>
CODE1	A 4 character alphanumeric variable which will contain the Unit Identification code. This code identifies the unit within the process. This value will be supplied by the RAM/MATRIX file. The variable will be used as an array.
CODE2	Same as above except the value is provided by the 94STOREA file.
TUNIT	This variable will be used to accumulate the total number of units which are identified by a specific code.
I, IJ, IK, IL	Various integer variables used as subscripts within the utility.

**12.2 OPERATING ENVIRONMENT:** This program is implemented on the EXECUTIVE-8 operating system.

**12.2.1 SUPPORT SOFTWARE:** - This utility requires the FORTRAN IV compiler and the facilities provided by the UNIVAC 1100/82 System.

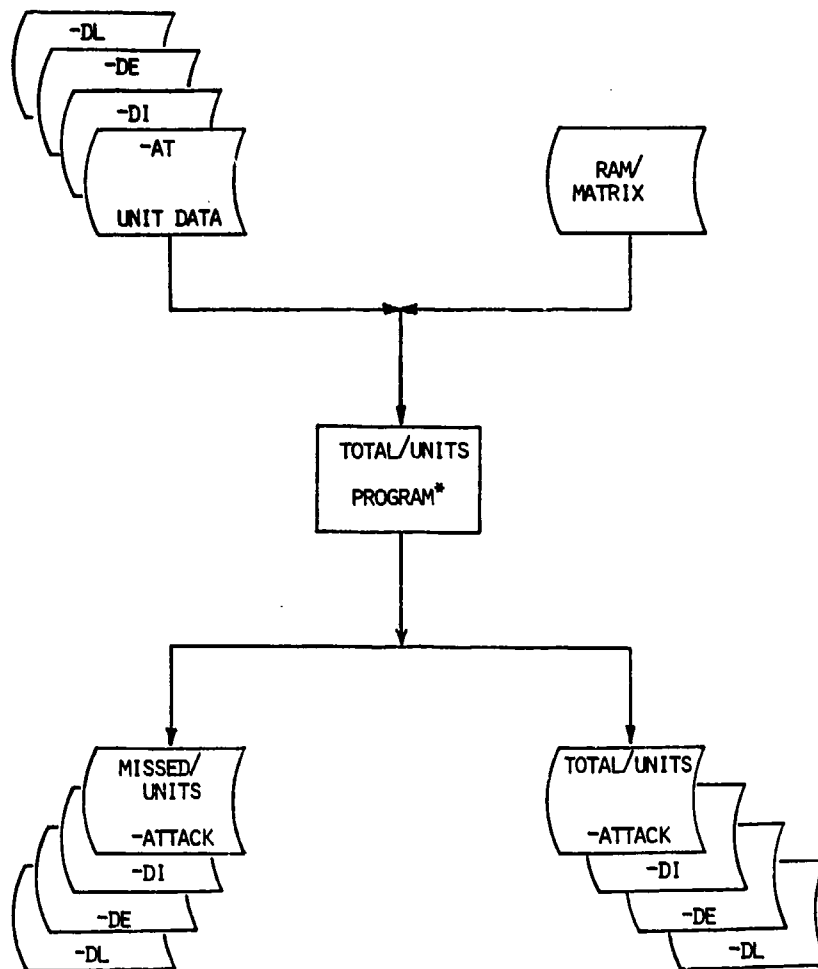
**12.2.2 I/O DEVICES:** - This utility will use as input files which reside on disk and will subsequently produce as output, files which will also reside on disk. Refer to Volume I for the program runstream.

**12.3 MAINTENANCE PROCEDURES:** This program has no specific maintenance procedures.

**12.3.1 PROGRAMMING CONVENTIONS:** - This a relatively short utility that is written in FORTRAN IV. As such, it has been programmed with standard FORTRAN programming techniques.

**12.3.2 INTERNAL ERROR ROUTINES:** - There are no explicit error handling routines written into this utility. However if unit type SRC codes in the RAM/MATRIX and stylized posture array file (94 STORE) do not agree the unit code from the stylized array file will be written out to the MISSED/UNITS - XX file.

# TOTAL/UNITS PROGRAM STRUCTURE



\* THE WIMP/TOTAL-UNITS MAY BE  
SUBSTITUTED FOR THIS PROGRAM

Figure III.12.1

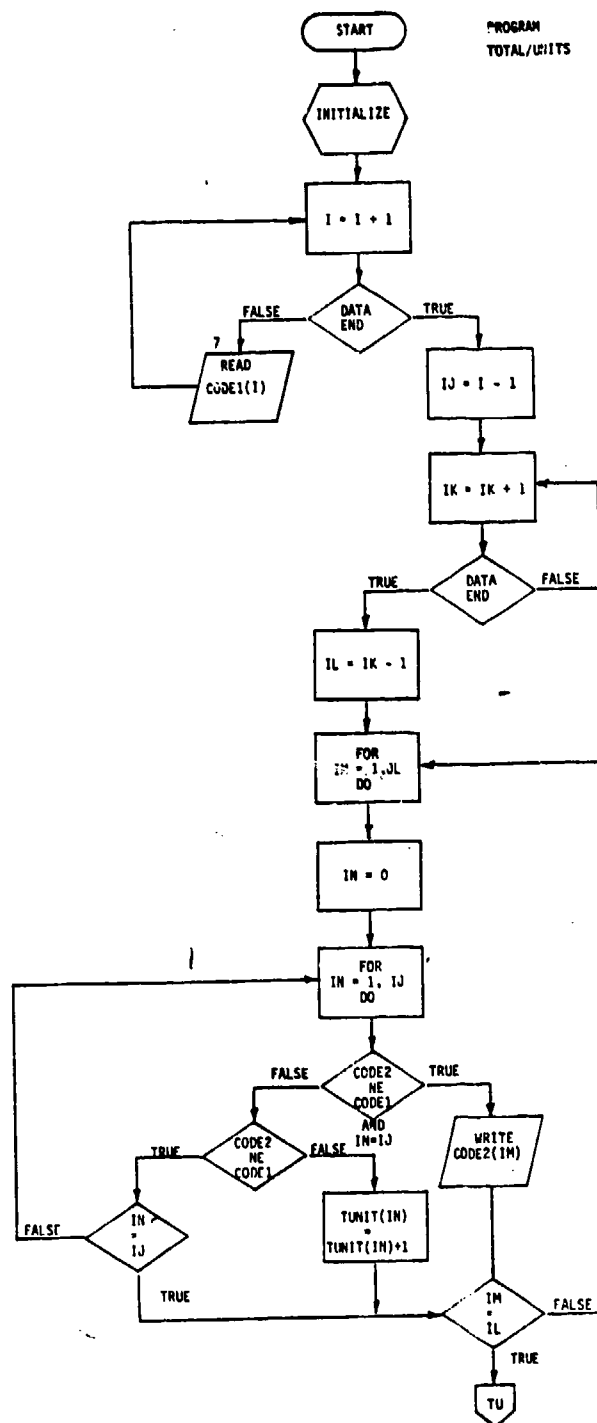


Figure III.12.2

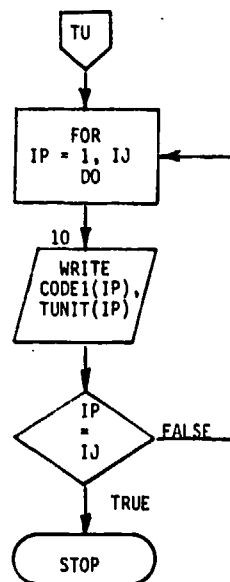


Figure III.12.2 (Cont)

UNCLASSIFIED\*\*\*FILE NAME:82XQT ELEMENT NAME:TOTAL/UNITS\*\*\*UNCLASSIFIED

```

1:C*****
2:C*****
3:C****THIS UTILITY IS DESIGNED TO TOTAL-UP THE NUMBER OF
4:C****UNITS BY TYPE (SRC) FROM THE FOUR STYLIZED POSTURE ARRAYS.
5:C****OUTPUT DATA FROM THIS UTILITY IS THEN USED IN UTILITY "82XQT.TOTAL
6:C****/CATEGORY", WHICH COMPUTES THE TOTAL EQUIPMENT IN EACH OF THE
7:C****22 ARTILLERY VULNERABILITY CATEGORIES WITHIN EACH STYLIZED
8:C****POSTURE ARRAY.
9:C*****
10:C*****
11:C****VARIABLE DICTIONARY
12:C****VARIABLE NAME      DEFINITION
13:C****CODE1(I)          UNIT TYPE CODE FROM 82RAMMATRIX.
14:C
15:C****CODE2(I)          UNIT CODES FROM STYLIZED ARRAY.
16:C
17:C****TUNIT(I)          TOTAL UNITS FOR EACH TYPE OF UNIT
18:C                      TOE.
19:C*****
20:C****READING INTO AN ARRAY TYPE UNIT CODES FROM THE 82RAM-
21:C****MATRIX DATA FILE.
22:C*****
23:      DIMENSION CODE1(500),CODE2(5000)
24:      INTEGER TUNIT(500)
25:10      I=1
26:      READ(7,700,END=99)CODE1(I)
27:700      FORMAT(1X,A4)
28:      GO TO 10
29:C*****
30:C****READING INTO AN ARRAY THE UNIT CODES FROM
31:C****THE STYLIZED ARRAY.
32:C*****
33:99      IJ=I-1
34:20      IK=IK+1
35:      READ(9,800,END=98)CODE2(IK)
36:800      FORMAT(70X,A4)
37:      GO TO 20
38:C*****
39:C****COMPUTING TOTAL NUMBER OF UNITS BY TYPE.
40:C*****
41:98      IL=IK-1
42:      DO 30 IM=1,IL
43:      IN=0
44:      DO 40 IN=1,IJ
45:      IF(CODE2(IM).NE.CODE1(IN).AND.IN.EQ.IJ)GO TO 45
46:      IF(CODE2(IM).NE.CODE1(IN))GO TO 40
47:      TUNIT(IN)=TUNIT(IN)+1
48:      GO TO 30
49:C*****
50:C****WRITING OUT ANY MISMATCHES BETWEEN ARRAY
51:C****AND TYPE UNIT LISTING.
52:C*****
53:45      WRITE(9,900)CODE2(IM)
54:900      FORMAT(A4)
55:      GO TO 30
56:40      CONTINUE
57:30      CONTINUE
58:C*****
59:C****WRITING OUT THE TOTAL NUMBER OF UNITS BY TYPE.
60:C*****
61:      DO 50 IP=1,IJ
62:      WRITE(10,1000)CODE1(IP),TUNIT(IP)
63:1000     FORMAT(A4,2X,I4)
64:50      CONTINUE
65:      STOP
66:      END

```

Figure III.12.3

UNCLASSIFIED\*\*\*FILE NAME:826WMP ELEMENT NAME:WMP/TOTAL-UNITS\*\*\*UNCLASSIFIED

```

1:C*****
2:C*****
3:C*****THIS UTILITY PROGRAM IS WRITTEN IN FORTRAN V.
4:C*****
5:C*****
6:C*****PROGRAMMER          DATE          INITIALS
7:C*****
8:C*****CPT SCOTT E CANTON   3FEB81       SEC   *
9:C*****
10:C*****
11:C*****
12:C*****PURPOSE OF THE PROGRAM:
13:C*****
14:C*****THIS UTILITY IS DESIGNED TO READ THE WMP INPUT FILE
15:C*****"UNITXXX", WHICH PROVIDES THE LISTING OF ALL BLUE AND
16:C*****RED UNITS ARRAYED IN THE POSTURE OF INTEREST, THE RAM/MATRIX
17:C*****DATA FILE AND TOTAL-UP ALL BLUE UNITS BY TOE TYPE.
18:C*****
19:C*****THIS UTILITY REPLACES TOTAL/UNITS, WHEN THE WMP IS USED IN
20:C*****PLACE OF WARP RAM.
21:C*****
22:C*****
23:C*****VARIABLE DICTIONARY
24:C*****
25:C*****VARIABLE NAME      DEFINITION
26:C*****CODEA(1)          UNIT TYPE ALPHA CODE DESCRIPTOR FOR EACH
27:C*****                  TYPE UNIT IN THE RAM/MATRIX DATA FILE.
28:C*****
29:C*****NCODE1(1)          UNIT TYPE CODE FROM RAM/MATRIX DATA
30:C*****                  FILE.
31:C*****
32:C*****NTYPE              UNIT CODE USED TO IDENTIFY BLUE UNITS
33:C*****                  FROM RED UNITS.
34:C*****
35:C*****CODE2(1)           UNIT CODE FROM COSAGE STYLIZED ARRAY UNIT
36:C*****                  FILE FOR A POSTURE.
37:C*****
38:C*****TUNIT(1)           TOTAL UNITS FOR EACH TYPE OF UNIT TOE.
39:C*****
40:C*****
41:C*****READING INTO ONE DIMENSION ARRAY TYPE OF UNIT CODES FROM
42:C*****RAM/MATRIX DATA FILE.
43:C*****
44:C*****
45:      DIMENSION CODEA(500),NCODE1(500),NCODE2(5000)
46:      INTEGER TUNIT(500)
47:10      II=II+1
48:      READ(7,700,END=20)CODEA(II),NCODE1(II)
49:700      FORMAT(1X,A2,I2)
50:      GO TO 10
51:C*****
52:C*****
53:C*****READING INTO AN ARRAY THE UNIT CODES FROM THE STYLIZED
54:C*****ARRAY FOR A POSTURE.
55:C*****
56:C*****
57:20      IK=IK+1

```

Figure III.12.4

UNCLASSIFIED\*\*\*FILE NAME:82WIMP ELEMENT NAME:WIMP/TOTAL-UNITS\*\*\*UNCLASSIFIED

```

50:21 READ(1,800,END=98)INTYPE,NCODE2(IK)
51:800 FORMAT(1X,13,2X,12)
60: IF(INTYPE.GE.800)GO TO 98
61: IF(NCODE2(IK).EQ.0)GO TO 21
62: GO TO 30
63:C*****
64:C*****
65:C*****COMPUTING TOTAL NUMBER OF UNITS BY TYPE CODE.
66:C*****
67:C*****
68:98 IL=IK-1
69: IJ=II-1
70: DO 30 I=1,IL
71: DO 40 J=1,IJ
72: IF(NCODE2(IP).NE.NCODE1(IN).AND.IN.EQ.IJ)GO TO 45
73: IF(NCODE2(IP).NE.NCODE1(IN))GO TO 40
74: TUNIT(IN)=TUNIT(IN)+1
75: GO TO 30
76:C*****
77:C*****
78:C*****WRITING OUT ANY MISMATCHES BETWEEN ARRAY AND RAM/MATRIX
79:C*****TYPE UNIT CODES LISTED.
80:C*****
81:C*****
82:45 WRITE(9,900)NCODE2(IM)
83:900 FORMAT(12)
84: GO TO 30
85:40 CONTINUE
86:30 CONTINUE
87:C*****
88:C*****
89:C*****WRITING OUT THE TOTAL NUMBER OF UNITS BY TCE TYPE.
90:C*****
91:C*****
92: DO 50 IF=1,IJ
93: WRITE(10,1000)CODEA(IP),NCODE1(IP),TUNIT(IP)
94:1000 FORMAT(A2,I2,2X,I4)
95:50 CONTINUE
96: STOP
97: END

```

Figure III.12.4 (Cont)

UNCLASSIFIED\*\*\*EXAMPLE OF THE OUTPUT DATA OF UTILITY RAW/MATRIX\*\*\*UNCLASSIFIED

1:	CH01	0	0	0	23	22	0	24	0	0	0	0	0	0	0	121	13	47	24	73	1	0	0	9	
2:	EN02	0	0	0	55	89	0	86	0	20	10	0	0	0	0	41142	79	315	271	101	147	20	0	15	
3:	EN03	0	0	0	24	32	0	24	0	7	2	0	0	0	0	4	202	9	55	71	43	11	7	0	15
4:	EN04	0	0	0	5	9	0	12	0	2	2	0	0	0	0	0	174	15	51	36	9	30	4	0	0
5:	EN05	0	0	0	11	20	0	14	0	5	0	0	0	0	0	0	150	6	56	56	22	16	3	0	0
6:	FA06	0	0	0	35	14	0	17	0	2	0	0	0	0	0	0	215	10	53	131	17	25	1	0	0
7:	FA07	0	19	0	43	39	0	24	0	5	0	2	54	2	045	74	130	356	78	118	3	0	0	0	
8:	FA08	0	0	0	29	3	0	13	0	1	0	0	0	0	0	240	17	51	234	43	73	2	0	0	
9:	FA09	0	0	0	3	6	0	2	0	1	0	0	15	0	108	17	22	34	8	13	0	0	0	0	
10:	FA10	0	0	0	5	19	0	5	0	1	0	2	9	2	31	6	19	20	11	6	1	0	0	0	
11:	FA11	0	19	0	59	54	0	37	0	21	1	2	48	0	757	82	170	416	115	106	7	0	0	0	
12:	FA12	0	1	0	34	4	0	15	0	1	0	0	0	0	234	70	61	224	53	58	1	0	0	0	
13:	FA13	0	4	0	4	8	0	3	0	1	0	0	12	0	105	14	21	37	8	10	1	0	0	0	
14:	FA14	0	6	0	7	3	0	4	0	1	0	0	0	0	44	11	26	53	25	11	1	0	0	0	
15:	FA15	0	0	0	6	23	0	9	0	16	1	2	12	0	174	7	20	23	13	7	2	0	0	0	
16:	FA16	0	12	0	42	42	0	28	0	5	1	2	42	0	544	57	123	210	53	64	1	0	0	0	
17:	FA17	0	0	0	27	3	0	16	0	1	0	0	0	0	145	11	41	93	19	21	0	0	0	0	
18:	FA18	0	4	0	3	7	0	2	0	1	0	0	11	0	104	13	22	32	8	12	0	0	0	0	
19:	FA19	0	0	0	6	18	0	6	0	1	1	2	9	0	47	7	16	21	10	7	1	0	0	0	
20:	FA20	0	30	0	78	74	0	40	0	58	0	2	0	0	542	13	92	321	101	23	1	0	0	0	
21:	FA21	0	0	0	24	11	0	19	0	1	0	2	0	0	125	4	19	75	20	14	1	0	0	0	
22:	FA22	0	10	0	18	21	0	7	0	19	0	0	0	0	139	5	21	82	25	3	0	0	0	0	
23:	ME23	0	13	0	34	31	0	46	0	5	0	6	0	2	976	142	459	452	43	87	3	0	0	0	
24:	ME24	0	0	0	14	23	0	19	0	5	0	2	0	2	184	21	43	74	29	15	3	0	0	0	
25:	ME25	0	3	0	3	2	0	5	0	0	0	1	0	0	203	35	93	95	3	12	0	0	0	0	
26:	ME26	0	1	0	1	0	0	1	0	0	0	0	0	0	78	11	30	28	0	2	0	0	0	0	
27:	ME27	0	4	0	11	2	0	12	0	0	0	1	0	0	158	16	137	93	5	10	0	0	0	0	
28:	ME28	0	0	0	69	35	0	58	0	8	0	0	0	0	400	0	58	80	50	16	4	0	0	0	
29:	ME29	0	0	0	27	14	0	28	0	2	0	0	0	0	154	0	22	41	23	4	1	0	0	0	
30:	MD30	0	0	0	14	7	0	10	0	2	0	0	0	0	32	0	12	13	9	4	1	0	0	0	
31:	MD31	0	0	0	6	21	0	10	0	2	0	0	0	0	126	0	16	34	16	2	62	0	1	0	
32:	MD32	0	0	0	45	2	0	6	0	1	0	0	0	0	102	0	15	22	13	3	0	0	0	0	
33:	MD33	0	0	0	9	9	0	17	0	2	0	0	0	0	190	0	72	241	14	57	2	0	0	0	
34:	OR34	0	0	0	8	40	0	13	0	6	3	0	0	0	261	4	21	59	25	5	4	0	0	0	
35:	OR35	0	0	0	11	43	0	26	0	3	0	0	0	0	199	12	17	53	16	10	0	0	0	0	
36:	CR36	0	0	0	2	35	0	20	0	13	0	0	0	1	171	9	47	44	23	3	8	0	0	0	
37:	AD37	0	0	0	11	23	0	10	0	3	0	0	0	0	129	4	14	38	13	1	7	0	0	0	
38:	AD38	0	0	0	23	17	0	16	0	6	0	0	0	0	119	3	12	44	10	2	11	0	0	0	
39:	GM39	0	0	0	9	14	0	12	0	6	0	0	0	0	123	4	12	18	10	4	0	6	0	0	
40:	CM40	0	0	0	11	24	0	13	0	5	0	0	0	0	147	3	23	99	28	5	2	2	0	0	
41:	SC41	0	0	0	162	45	0	59	0	4	0	0	0	0	709	10	63	952	199	593	9	0	0	0	
42:	SC42	0	0	0	13	18	0	19	0	1	0	0	0	0	104	1	17	109	24	3	9	0	0	0	
43:	SC43	0	0	0	52	10	0	17	0	1	0	0	0	0	274	4	16	352	60	224	0	0	0	0	
44:	SC44	0	0	0	47	5	0	8	0	1	0	0	0	0	197	4	15	217	45	132	0	0	0	0	
45:	SC45	0	0	0	50	12	0	15	0	1	0	0	0	0	189	1	15	274	50	339	0	0	0	0	
46:	AC46	0	0	0	7	4	0	8	0	2	0	0	0	0	285	4	7	17	7	2	1	0	0	0	
47:	FC47	0	0	0	2	2	0	3	0	0	0	0	0	0	34	3	7	13	6	1	0	0	0	0	
48:	AR48	0	0	0	30	5	0	27	0	1	0	0	0	0	191	4	29	60	29	1	2	0	0	0	
49:	AR49	0	4	2	44	37	0	47	0	5	5	2	0	0	644	24	249	286	48	69	3	0	0	0	
50:	AR50	0	0	0	20	29	0	25	0	5	2	1	0	0	193	2	45	101	12	16	3	0	0	0	
51:	AR51	0	0	0	5	2	0	4	0	0	1	0	0	0	129	1	40	15	4	9	0	0	0	0	
52:	AR52	0	0	0	1	0	0	0	0	0	0	0	0	0	42	0	10	4	0	2	0	0	0	0	
53:	AR53	0	4	2	9	2	0	10	0	0	0	1	0	0	107	19	94	68	4	26	0	0	0	0	
54:	CA54	0	0	0	19	4	0	14	0	1	0	1	0	0	107	6	19	67	26	3	2	0	0	0	
55:	CA55	0	9	0	44	46	0	43	0	6	3	2	2	31039	109	540	506	70	183	6	2	0	0	0	
56:	CA56	0	0	0	26	24	0	19	0	5	0	2	0	3	208	33	47	49	27	23	3	0	0	0	
57:	CA57	0	0	0	6	16	0	15	0	1	0	0	2	0	225	13	107	199	19	64	1	2	0	0	

Figure III.12.5

[illegible]

285

UNCLASSIFIED\*\*\*EXAMPLE OF OUTPUT FROM UTILITY TOTAL/UNIT

1:CH23  
2:AR35  
3:IN42  
4:AH57

Figure III.12.7

UNCLASSIFIED\*\*\*EXAMPLE OF OUTPUT OF UTILITY TOTAL/UNITS\*\*\*UNCLASSIFIED

1:1L197	2
2:1L199	1
3:1L241	1
4:1L242	1
5:1L303	1
6:1L243	1
7:1L427	4
8:1SC21	1
9:1SC28	1
10:1SC22	1
11:1SC23	1
12:1SC24	1
13:1SC25	1
14:1SC26	1
15:1SC25	4
16:1SC26	4
17:1SC27	5
18:1SC28	4
19:1T011	7
20:1T011	28
21:1T011	84
22:1MF16	4
23:1MC17	8
24:1MC17	24
25:1MT17	8
26:1RC06	8
27:1R939	4
28:1ML24	4
29:1MF34	12
30:1M134	4
31:1ML72	1
32:1L198	1
33:1L207	4
34:1AH47	1
35:1A047	2
36:1A747	2
37:1AD79	1
38:1AD80	1
39:1AD81	2
40:1AN19	4
41:1L291	1
42:1C241	1
43:1L491	8
44:1C593	1
45:1L494	1
46:1L495	1
47:1L496	1
48:1MS12	4
49:1MS13	4
50:1MS14	4
51:1MS15	4
52:1RW33	1
53:1CA7C	4
54:1CM59	1
55:1CHE0	1
56:1CM58	1
57:1AD46	2

Figure III.12.8

UNCLASSIFIED\*\*\*EXAMPLE OF THE OUTPUT OF UTILITY WIMP/TOTAL-UNITS\*\*\*UNCLASSIFIED

1:FA14	1
2:FA15	9
3:FA16	10
4:FA19	1
5:FA17	7
6:FA18	3
7:ME 2	3
8:ME 3	4
9:ME 4	15
10:ME 5	32
11:FA 6	1
12:ME 7	3
13:ME 8	11
14:AR 1	1
15:AR 9	6
16:AR10	14
17:AR11	41
18:AR12	4
19:AR13	12
20:AH20	3

Figure III.12.9

## CHAPTER 13

### UTILITY - TOTAL/CATEGORY

#### AND

### WIMP/TOTAL - CATEGORY

**13.1 DESCRIPTION OF PROCESSING:** This program sums the categories of equipment in the force array. Otherwise logical read and write statements perform the program functions.

**13.1.1 PURPOSE/FUNCTIONS:** The purpose of this utility is to produce the total authorized quantity of equipment for each of the 22 artillery vulnerability categories found in one of the four stylized posture arrays. In accomplishing this function the utility is run once for each of the posture arrays using the RAM/MATRIX file and the appropriate TOTAL/UNIT-XX file. The utility will first multiply the total number of units in a unit type as specified in the TOTAL/UNIT-XX file by the number of items of equipment in a particular vulnerability category as specified in the RAM/MATRIX file. The final step in this process is to determine the total number of items of equipment in each of the 22 vulnerability categories by summing each of the categories across all unit types. The result will be an array, 22 entries in length, one entry for each artillery vulnerability category and the total number of items of equipment authorized for that category for a specific posture.

**13.1.2 PROGRAM INPUT/OUTPUT STRUCTURE:** The structure of the utility is displayed in FIGURE III.13.1. The logic flow found in the utility is shown in FIGURE III.3.2. The source code for the utility is listed in FIGURE III.13.3; Figure III.13.4 is the source code for the alternate version of the program, WIMP/TOTAL-CATEGORY.

**13.1.2.A INPUT DATA AND DATA BASE:** There are two input files required to run this utility. Both files will be cataloged as elements under the general file for the current study. The first file is the RAM/MATRIX. This file details for each unit in the study the number of items of equipment that the unit has been authorized classified in the 22 artillery vulnerability categories. The second file used as input will be one of the TOTAL/UNIT-XX files. This file will denote the total number of unit types found in the particular posture array.

Figure III.13.5 depicts the record layout for the RAM/MATRIX file and examples of its data. Figure III.13.6 depicts the record layout and data examples for the TOTAL/UNITS-XX file.

**13.1.2.B OUTPUT DATA AND DATA FILES:** The TOTAL/CATEGORY-XX file is the only output file from each run of this utility. As with the TOTAL/UNITS utility the TOTAL/CATEGORY utility must be run four times; once for each of the four posture arrays of the study.

The output file will have one record type and be limited to 22 records; one for each of the 22 artillery vulnerability categories of the study. Each record will

consist of one seven-digit integer field. This field will contain the total number of items of equipment found in a specific posture array for a single vulnerability category. For example in Figure III.13.7 line 5 indicates that 4,060 items of equipment are being played in this posture which have been assigned an artillery vulnerability category of 5.

These 22 totals in this posture will become the denominators of the equation which will be used to determine the WARFRAM loss rates within a posture's stylized array. The RAM(WARF) simulation runs for each posture will provide the number of equipment items lost by vulnerability category. These figures will be used as the numerators in the equation. These loss rates will be manually computed and entered into the element "CONTROL/TEMP" of the current study's general file and be used as an input file to a subsequent utility.

Figures III.13.8 and presents the file layout and data examples for the WIMP/TOTAL-CAT-XX.

### 13.1.2.C DATA ELEMENT DICTIONARY:

The following section identifies and defines all variables used in this utility.

<u>NAME</u>	<u>DEFINITION</u>
CAT	An integer variable which is established as an array of 22 occurrences; one occurrence for each of the 22 artillery vulnerability categories. In each occurrence is accumulated the total authorized quantity of equipments in the study which have been assigned this vulnerability classification.
VULCAT	This is an integer variable which is established as a 2 dimensional array within the utility. The first dimension of the array will vary for each type of equipment in the study and that was identified in the LINCDEF/-LIST file. The second dimension will identify the artillery vulnerability category which this type of equipment has been assigned. Into the appropriate category for a particular type of equipment will be entered the total authorized quantity of equipment. These quantities are then multiplied by the total number of units to determine the total number of items being played.
TUNITS	This is an integer variable which identifies the total number of units of a specific type being played. Since this utility is calculating the total number of items in each vulnerability category no attention is paid to the type of unit being processed. This data is supplied by the TOTAL/UNITS file.
K, KK, L, JJ, KM, LN, LL, NN	Various integer variables used as subscripts by the utility.

13.2 OPERATING ENVIRONMENT: This program is implemented on the EXECUTIVE-8 operating system.

13.2.1 SUPPORT SOFTWARE : This utility requires the FORTRAN IV compiler and the UNIVAC 1100 system facilities. Refer to Volume I for the program runstream.

13.2.2 I/O DEVICES : This utility uses as input, files which reside on disk and produces as output files which will also reside on disk.

13.3 MAINTENANCE PROCEDURES: There are no special maintenance procedures developed.

13.3.1 PROGRAMMING CONVENTIONS : This is a relatively compact utility written in FORTRAN IV and follows those normally found in FORTRAN programming.

13.3.2 INTERNAL ERROR ROUTINES: The utility has no explicit error routines written into it. As a result only errors detected by the system and the appropriate error messages will be provided in order to correct problems.

# TOTAL/CATEGORIES STRUCTURE

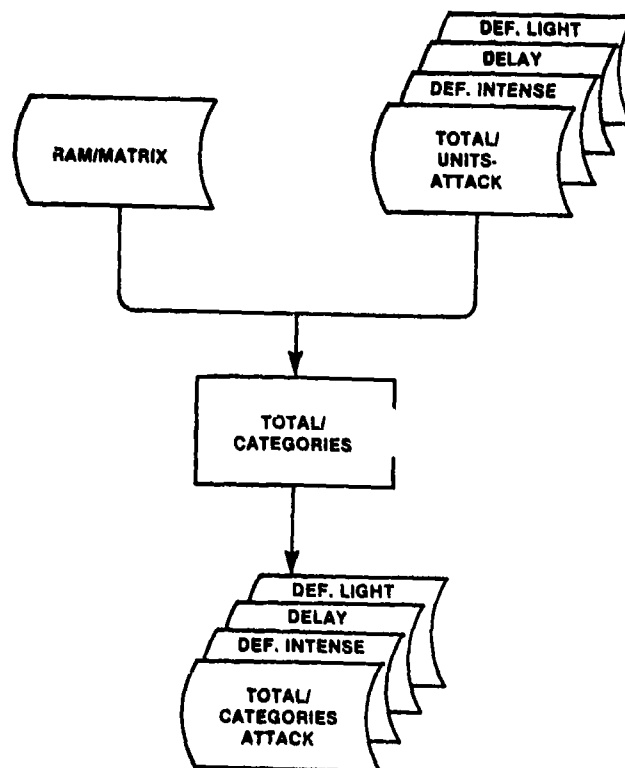


Figure III.13.1

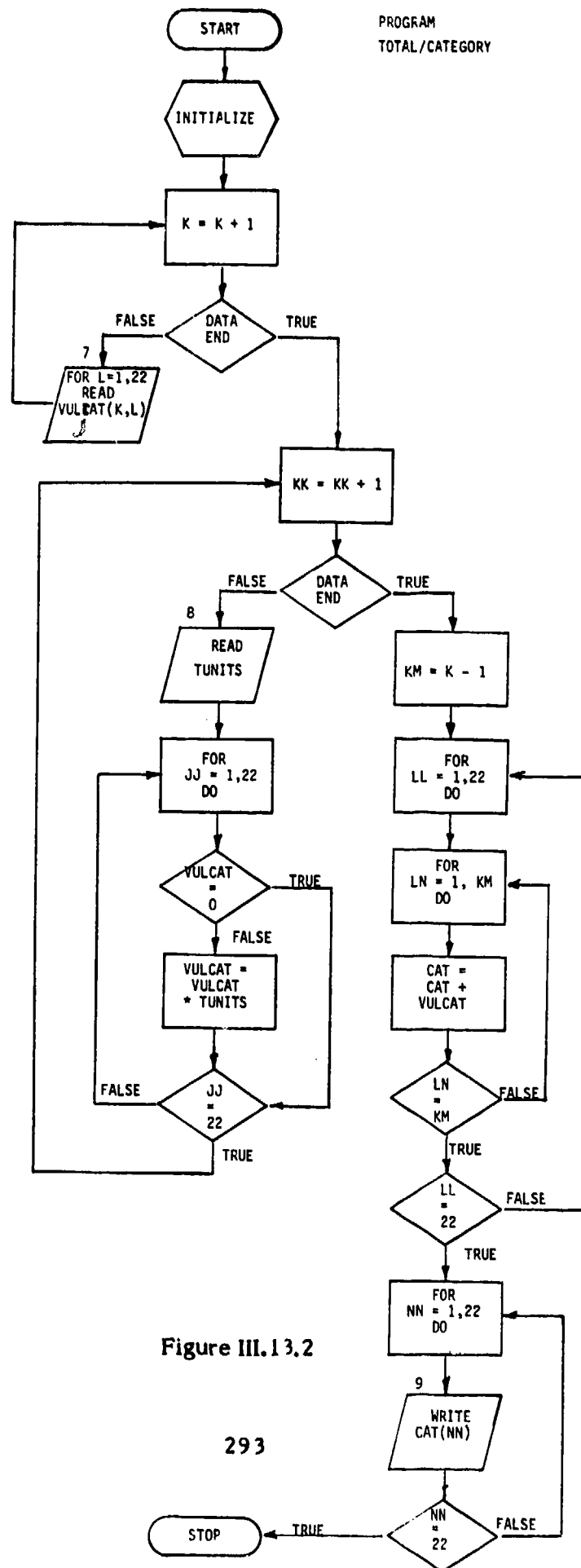


Figure III.13.2

UNCLASSIFIED\*\*FILE NAME:82XQT ELEMENT NAME:TOTAL/CATEGORY\*\*UNCLASSIFIED

```

1:C*****
2:C*****
3:C*****THIS PROGRAM IS WRITTEN IN FORTRAN IV.
4:C*****
5:C*****
6:C*****PURPOSE OF PROGRAM:
7:C*****
8:C*****THIS UTILITY PRODUCES THE TOTAL AUTHORIZED QUANTITY OF
9:C*****EQUIPMENT FOR EACH OF THE 22 ARTILLERY VULNERABILITY CATEGORIES
10:C*****FOUND IN A POSTURE'S STYLIZED ARRAY.
11:C*****THE OUTPUT OF UTILITY "82XQT.TOTAL/UNITS", WHICH PROVIDES
12:C*****THE TOTAL NUMBER OF EACH TYPE OF UNIT IN A STYLIZED ARRAY
13:C*****IS MULTIPLIED TIMES THE TOTAL QUANTITY OF EQUIPMENT
14:C*****AUTHORIZED IN EACH OF THE 22 ARTILLERY VULNERABILITY CATEGORIES
15:C*****FOR EACH TYPE OF UNIT FOUND IN THE POSTURE'S STYLIZED ARRAY.
16:C*****THE OUTPUT OF THIS UTILITY IS A TOTAL QUANTITY OF EQUIPMENT
17:C*****FOUND IN A POSTURE'S STYLIZED ARRAY FOR EACH OF THE 22
18:C*****ARTILLERY VULNERABILITY CATEGORIES.
19:C*****THESE TOTALS THEN BECOME THE DENOMINATORS IN DETERMINING THE
20:C*****WARFRAM LOSS RATES FOR EACH OF THE 22 ARTILLERY VULNERABILITY
21:C*****CATEGORIES WITH IN A POSTURE'S STYLIZED ARRAY.
22:C*****THE RAM (WARF) SIMULATION RUNS FOR EACH POSTURE WILL PROVIDE
23:C*****THE NUMBER OF ITEMS OF EQUIPMENT LOST BY CATEGORY, WHICH WILL
24:C*****SERVE AS THE NUMERATOR IN COMPUTING THE WARFRAM LOSS RATES IN
25:C*****EACH CATEGORY FOR AN ARRAYED POSTURE. THE 88 WARFRAM LOSS RATES
26:C*****ARE COMPUTED MANUALLY AND EDITED MANUALLY INTO THE
27:C*****ELEMENT "CONTROL/TEMP" OF THE WARF STUDY'S PROGRAM FILE.
28:C*****
29:C*****
30:C*****VARIABLE DICTIONARY
31:C*****VARIABLE NAME      DEFINITION
32:C*****VULCAT(I,J)      AUTHORIZED TOE EQUIPMENT COMPATABLE
33:C                        WITH WARF STUDY'S LIN CODE LIST
34:C                        IN EACH VULNERABILITY CATEGORY.
35:C
36:C*****TUNITS            TOTAL NUMBER OF UNITS BY TYPE
37:C                        OF UNIT TOE.
38:C
39:C*****CAT(I)            TOTAL AUTHORIZED TOE EQUIPMENT
40:C                        BY VULNERABILITY CATEGORY FOR
41:C                        THE ENTIRE ARRAYED POSTURE.
42:C*****
43:C*****
44:      INTEGER VULCAT(500,22),CAT(23)
45:C*****
46:C*****READING IN AUTHORIZED EQUIPMENT BY TYPE UNIT/VULNERABILITY
47:C*****CATEGORY.
48:C*****
49:10      K=K+1
50:      READ(7,700,END=99)(VULCAT(K,L),L=1,22)
51:700      FORMAT(6X,22I4)
52:      GO TO 10
53:C*****
54:C*****READING IN TOTAL NUMBER OF UNITS BY TYPE IN THE ARRAY.
55:C*****
56:99      KK=KK+1
57:      READ(8,800,END=97)TUNITS

```

Figure III.13.3

UNCLASSIFIED\*\*\*FILE NAME:82XQT ELEMENT NAME:TOTAL/CATEGORY\*\*\*UNCLASSIFIED

```
58:800  FORMAT(6X,I4)
59:C*****
60:C*****MULTIPLYING TOTAL NUMBER OF UNITS BY TYPE TIMES
61:C*****AUTHORIZED EQUIPMENT PER VULNERABILITY CATEGORY.
62:C*****
63:      DO 20 JJ=1,22
64:      IF(VULCAT(KK,JJ).EQ.0)GO TO 20
65:      VULCAT(KK,JJ)=VULCAT(KK,JJ)*TUNITS
66:20    CONTINUE
67:      GO TO 99
68:C*****
69:C*****TOTALING UP VULNERABILITY CATEGORY FOR ALL UNITS IN
70:C*****THE ARRAY.
71:C*****
72:98    KM=K-1
73:      DO 30 LL=1,22
74:      GO 40 LN=1,KM
75:      CAT(LL)=CAT(LL)+VULCAT(LN,LL)
76:40    CONTINUE
77:30    CONTINUE
78:C*****
79:C*****WRITING OUT TOTALS BY VULNERABILITY CATEGORY FOR THE
80:C*****ARRAY.
81:C*****
82:      DO 50 NN=1,22
83:      WRITE(9,90)CAT(NN)
84:900    FORMAT(I7)
85:50    CONTINUE
86:      STOP
87:      END
```

Figure III.13.3 (Cont)

UNCLASSIFIED\*\*\*FILE NAME:W2XCT ELEMENT NAME:WIMP/TOTAL-CAT\*\*\*UNCLASSIFIED

```

1:C*****
2:C*****
3:C*****THIS UTILITY PROGRAM IS WRITTEN IN FORTRAN V.
4:C*****
5:C*****
6:C*****PROGRAMMER          DATA          INITIALS
7:C*****
8:C*****CPT SCOTT E CARTLON          4FEBP1          SEC
9:C*****
10:C*****
11:C*****
12:C*****PURPOSE OF PROGRAM:
13:C*****
14:C*****THIS UTILITY IS DESIGNED TO TOTAL THE QUANTITIES OF EQUIPMENT
15:C*****AUTHORIZED ALL BLUE UNITS WITH IN AN ARRAYED POSTURE BY
16:C*****THE 22 ARTILLERY VULNERABILITY CATEGORIES. THE FIRST
17:C*****CATEGORY FOR PERSONNEL IS SET TO ZERO AND NOT EFFECTED
18:C*****BY THIS UTILITY.
19:C*****THIS UTILITY USES INPUT FROM TWO UTILITIES "WIMP/MATRIX"
20:C*****AND "WIMP/TOTAL-UNITS". THE OUTPUT DATA CREATED BY THIS
21:C*****UTILITY IS USED BY THE UTILITY "WIMP/LOSS-RATES" TO COMPUTE
22:C*****THE BLUE LOSSES BY ARTILLERY VULNERABILITY CATEGORY TO RED
23:C*****ARTILLERY FOR A POSTURE. THIS UTILITY MUST BE RUN ONCE
24:C*****FOR EACH STYLIZED POSTURE ARRAY (IE. ATTACK, DEFENSE INTENSE,
25:C*****DELAY AND OFFENSE LIGHT.
26:C*****
27:C*****THIS UTILITY REPLACES UTILITY "TOTAL/CATEGORY" WHEN THE
28:C*****WIMP IS USED INPLACE OF WARF RAM.
29:C*****
30:C*****
31:C*****VARIABLE DICTIONARY
32:C*****VARIABLE NAME      DEFINITION
33:C*****
34:C*****VULCAT(I,J)        AUTHORIZED TQE EQUIPMENT PLACED IN
35:C*****                    THE 23 PERSONEL/EQUIPMENT ARTILLERY VULNER-
36:C*****                    ABILITY CATEGORIES FOR EACH TYPE
37:C*****                    OF UNIT.
38:C*****
39:C*****TUNITS              TOTAL NUMBER OF UNITS BY TYPE OF
40:C*****                    TQE.
41:C*****
42:C*****NPFERS              DUMMY VARIABLE USED TO SET CATEGORY
43:C*****                    ONE (PERSONNEL) TO ZERO.
44:C*****
45:C*****CAT(I)              TOTAL AUTHORIZED TQE EQUIPMENT BY
46:C*****                    ARTILLERY VULNERABILITY CATEGORY
47:C*****                    FOR ALL BLUE UNITS IN THE STYLIZED
48:C*****                    POSTURE ARRAY.
49:C*****
50:C*****
51:C*****
52:      INTEGER VULCAT(500,23),CAT(23),TUNITS
53:C*****
54:C*****
55:C*****READING IN THE AUTHORIZED EQUIPMENT BY TYPE UNIT
56:C*****AND VULNERABILITY CATEGORY.
57:C*****

```

Figure III.13.4

UNCLASSIFIED\*\*\*FILE NAME:60XCT ELEMENT NAME:IMP/TOTAL-CAT\*\*\*UNCLASSIFIED

```

56:C*****
59:10      KK=1
60:      READ(7,700,FND=99)(VULCAT(K,L),L=1,22)
61:700     FORMAT(17X,2D14)
62:      GO TO 10
63:C*****
64:C*****
65:C*****READING IN TOTAL NUMBER OF UNITS BY TYPE IN THE ARRAY.
66:C*****
67:C*****
68:99      KK=KK+1
69:      READ(8,800,FND=96)TUNITS
70:800     FORMAT(6X,14)
71:C*****
72:C*****
73:C*****MULTIPLYING THE TOTAL NUMBER OF UNITS BY TYPE TIMES
74:C*****THE AUTHORIZED QUANTITY OF EQUIPMENT FOR EACH VULNER-
75:C*****ABILITY CATEGORY.
76:C*****
77:C*****
78:      DO 20 JJ=1,22
79:      IF(VULCAT(KK,JJ).EQ.0)GO TO 20
80:      VULCAT(KK,JJ)=VULCAT(KK,JJ)*TUNITS
81:20      CONTINUE
82:      GO TO 99
83:C*****
84:C*****
85:C*****TOTALING UP THE EQUIPMENT BY VULNERABILITY CATEGORY
86:C*****FOR ALL UNITS IN THE STYLIZED POSTURE ARRAY.
87:C*****
88:C*****
89:C*****
90:98      KK=K-1
91:      DO 30 LL=1,22
92:      DO 40 LN=1,KN
93:      CAT(LL)=CAT(LL)+VULCAT(LN,LL)
94:40      CONTINUE
95:30      CONTINUE
96:C*****
97:C*****
98:C*****FITTING OUT THE ZERO VALUE IN CATEGORY ONE (PERSONNEL).
99:C*****
100:C*****
101:      NPERS=0
102:      WRITE(9,900)NPERS
103:900      FORMAT(17)
104:C*****
105:C*****
106:C*****WRITING OUT EQUIPMENT TOTALS FOR VULNERABILITY CATEGORIES
107:C*****THO THROUGH TWENTY-THREE FOR THE ENTIRE ARRAY.
108:C*****
109:C*****
110:      DO 50 NN=1,22
111:      WRITE(9,900)(CAT(NN))
112:50      CONTINUE
113:      STOP
114:      END

```

Figure III.13.4 (Cont)

UNCLASSIFIED\*\*\*EXAMPLE OF OUTPUT OF UTILITY TOTAL/UNITS\*\*\*UNCLASSIFIED

1:LL197	2
2:LL199	1
3:LL2A1	1
4:LL2A2	1
5:LL303	1
6:LL2A3	1
7:LL427	4
8:SSC51	1
9:SSC50	1
10:SSC52	1
11:SSC53	1
12:SSC54	1
13:SSC55	1
14:SSC56	1
15:SSC35	4
16:SSC36	4
17:SSC37	5
18:SSC38	4
19:TP11	7
20:TC11	28
21:TP11	84
22:MF16	4
23:MC17	8
24:ME17	24
25:MT17	8
26:RC06	8
27:R939	4
28:ML34	4
29:MF34	12
30:MI34	4
31:ML72	1
32:LL198	1
33:LL207	4
34:AM47	1
35:AB47	2
36:AT47	2
37:AD79	1
38:AD80	1
39:AD81	2
40:AW19	4
41:LL291	1
42:CP41	1
43:LL492	8
44:CS93	1
45:LL494	1
46:LL495	1
47:LL496	1
48:MS12	4
49:MS13	4
50:MS14	4
51:MS15	4
52:RW33	4
53:CM30	4
54:CH59	1
55:CH60	1
56:CH58	1
57:AD46	2

Figure III.13.5

## UNCLASSIFIED\*\*\*EXAMPLE OF THE OUTPUT DATA OF UTILITY RAM/MATRIX\*\*\*UNCLASSIFIED

1: CH01	0	0	0	23	22	0	24	0	0	0	0	0	121	13	37	24	73	1	0	0	0
2: EN02	0	0	4	55	88	0	86	0	20	10	0	0	41142	79	315	271	101	147	20	0	15
3: EN03	0	0	0	24	32	0	24	0	7	2	0	0	4	202	9	51	71	43	11	7	0
4: EN04	0	0	0	5	9	0	12	0	2	2	0	0	0	194	15	51	36	9	30	4	0
5: EN05	0	0	4	11	20	0	14	0	5	0	0	0	0	150	6	50	56	22	16	3	0
6: FA06	0	0	0	35	14	0	17	0	2	0	0	0	0	215	10	52	131	17	25	1	0
7: FA07	0	18	0	43	39	0	24	0	5	0	2	54	2	045	74	131	350	78	118	3	0
8: FA08	0	0	0	29	3	0	13	0	1	0	0	0	0	240	17	51	234	43	73	2	0
9: FA09	0	0	0	3	6	0	2	0	1	0	0	15	0	108	17	22	34	8	13	0	0
10: FA10	0	0	0	5	18	0	5	0	1	0	2	9	2	91	8	19	20	11	6	1	0
11: FA11	0	19	0	59	54	0	37	0	21	1	2	48	0	767	82	170	410	115	106	7	0
12: FA12	0	1	0	34	4	0	15	0	1	0	0	0	0	234	70	61	224	53	58	1	0
13: FA13	0	4	0	4	8	0	3	0	1	0	0	12	0	105	14	21	37	8	10	1	0
14: FA14	0	6	0	7	3	0	4	0	1	0	0	0	0	94	11	20	53	25	11	1	0
15: FA15	0	0	0	0	23	0	9	0	16	1	2	12	0	134	9	20	23	13	7	2	0
16: FA16	0	12	0	42	42	0	28	0	5	1	2	42	0	544	57	123	210	53	64	1	0
17: FA17	0	0	0	27	3	0	16	0	1	0	0	0	0	145	11	41	93	19	21	0	0
18: FA18	0	4	0	3	7	0	2	0	1	0	0	11	0	104	13	22	32	8	12	0	0
19: FA19	0	0	0	6	18	0	6	0	1	1	2	9	0	97	7	16	21	10	7	1	0
20: FA20	0	30	0	78	74	0	40	0	58	0	2	0	0	542	19	32	321	101	23	1	0
21: FA21	0	0	0	24	11	0	19	0	1	0	2	0	0	125	4	10	75	20	14	1	0
22: FA22	0	10	0	16	21	0	7	0	19	0	0	0	0	139	5	21	82	25	3	0	0
23: ME23	0	13	0	34	31	0	46	0	5	0	6	0	2	970	142	459	452	43	87	3	0
24: ME24	0	0	0	14	23	0	19	0	5	0	2	0	2	184	21	43	74	29	15	3	0
25: ME25	0	3	0	3	2	0	5	0	0	0	1	0	0	208	35	93	95	3	12	0	0
26: ME26	0	1	0	1	0	0	1	0	0	0	0	0	0	64	11	30	28	0	2	0	0
27: ME27	0	4	0	11	2	0	12	0	0	0	1	0	0	168	10	137	93	5	10	0	0
28: MD28	0	0	0	69	35	0	58	0	8	0	0	0	0	400	0	58	80	50	16	4	0
29: MD29	0	0	0	27	14	0	28	0	2	0	0	0	0	154	0	22	41	23	4	1	0
30: MD30	0	0	0	14	7	0	10	0	2	0	0	0	0	82	0	12	13	9	4	1	0
31: MD31	0	0	0	6	21	0	10	0	2	0	0	0	0	126	0	16	34	16	2	62	0
32: MD32	0	0	0	45	2	0	6	0	1	0	0	0	0	102	0	15	22	13	3	0	0
33: MD33	0	0	0	9	9	0	17	0	2	0	0	0	0	190	0	72	241	34	57	2	0
34: OR34	0	0	0	8	40	0	13	0	6	0	0	0	0	261	4	21	59	25	5	4	0
35: OR35	0	0	0	11	43	0	26	0	3	0	0	0	0	199	12	17	59	16	10	0	0
36: CR36	0	0	0	7	35	0	20	0	13	0	0	0	1	171	9	47	44	23	3	6	0
37: AD37	0	0	0	11	23	0	10	0	3	0	0	0	0	129	4	14	38	13	1	7	0
38: AD38	0	0	0	23	17	0	16	0	6	0	0	0	0	119	3	12	44	10	2	11	0
39: GM39	0	0	0	9	14	0	12	0	6	0	0	0	0	129	4	12	18	10	4	0	6
40: GM40	0	0	0	11	24	0	13	0	5	0	0	0	0	197	3	23	89	28	5	2	2
41: SC41	0	0	0	162	45	0	59	0	4	0	0	0	0	709	10	63	952	189	698	9	0
42: SC42	0	0	0	13	18	0	19	0	1	0	0	0	0	104	1	17	109	24	3	9	0
43: SC43	0	0	0	52	10	0	17	0	1	0	0	0	0	234	4	16	352	60	224	0	0
44: SC44	0	0	0	47	5	0	8	0	1	0	0	0	0	192	4	15	217	45	132	0	0
45: SC45	0	0	0	50	12	0	15	0	1	0	0	0	0	189	1	15	274	50	339	0	0
46: AG46	0	0	0	7	4	0	8	0	2	0	0	0	0	292	4	7	17	7	2	1	0
47: FC47	0	0	0	2	2	0	3	0	0	0	0	0	0	94	3	7	13	6	1	0	0
48: AR48	0	0	0	30	5	0	27	0	1	0	0	0	0	191	4	29	60	29	1	2	0
49: AR49	0	4	2	44	37	0	47	0	5	5	2	0	0	644	24	249	286	48	69	3	0
50: AR50	0	0	0	20	29	0	25	0	5	2	1	0	0	193	2	45	101	32	16	3	0
51: AR51	0	0	0	5	2	0	4	0	0	1	0	0	0	128	1	40	39	4	9	0	0
52: AR52	0	0	0	1	0	0	0	0	0	0	0	0	0	42	0	10	8	0	2	0	0
53: AR53	0	4	2	9	2	0	10	0	0	0	1	0	0	107	19	84	68	4	26	0	0
54: CA54	0	0	0	19	4	0	14	0	1	0	1	0	0	107	6	19	67	26	3	2	0
55: CA55	0	9	0	44	46	0	43	0	6	3	2	2	31039	109	540	506	70	183	6	2	0
56: CA56	0	0	0	26	24	0	19	0	5	0	2	0	3	208	39	47	89	27	23	3	0
57: CA57	0	0	0	6	16	0	15	0	1	0	0	2	0	225	13	103	198	19	64	3	2

Figure III.13.6

UNCLASSIFIED\*\*\*EXAMPLE OF OUTPUT OF UTILITY TOTAL/CATEGORY\*\*\*UNCLASSIFIED

1:	0
2:	166
3:	0
4:	3457
5:	4060
6:	0
7:	1333
8:	342
9:	259
10:	50
11:	28
12:	253
13:	71
14:	3093
15:	4296
16:	855
17:	3667
18:	1049
19:	78483
20:	750
21:	0
22:	3

Figure III.13.7

UNCLASSIFIED\*\*\*EXAMPLE OF THE OUTPUT DATA OF UTILITY LIPP/TOTAL-CAT\*\*\*UNCLASSIFIED

1:	0
2:	336
3:	251
4:	0
5:	1051
6:	718
7:	0
8:	952
9:	6
10:	120
11:	30
12:	37
13:	175
14:	170
15:	27316
16:	2914
17:	17042
18:	14101
19:	1786
20:	7075
21:	91
22:	57
23:	0

Figure III.13.8

**THIS PAGE IS INTENTIONALLY LEFT BLANK**

## CHAPTER 14

### UTILITY - SEARCH/ENGAGEREP

**14.1 DESCRIPTION OF PROCESSING:** The program performs multiple computations on the data based upon combat posture and scenario time periods. The program makes calls to two subroutines.

**14.1.1 PURPOSE/FUNCTIONS:** The purpose of this utility is to determine the fraction of time spent by the U.S. Force in each of the four combat postures in each of the seven time periods of the study. The four combat postures are Attack, Defend (Defense Intense), Delay and Defense Light (Inactive). The seven time periods of the study are:

	<u>PERIODS</u>						
	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>	<u>6</u>	<u>7</u>
DAYS	1-15	16-30	31-60	61-90	91-120	121-150	151-180

In order to determine this fraction of time the utility uses as its only input, the Engagement Report, which is produced by the Concepts Evaluation Model (CEM) theater simulator. This file may either be found on magnetic tape or on mass storage. Depending on the location of the file, a different runstream will be used to execute the utility.

The output from this utility will be a mass storage file "SCENARIO-XX" which will be used as input to the following utility "CONTROL/COMPILER".

**14.1.2 PROGRAM INPUT/OUTPUT STRUCTURE:** The overall structure of the utility is displayed in Figure III.14.1. The flow of logic found in the utility is pictured in Figure III.14.2 and Figure.14.2 A. The source code of the utility is listed in FIGURE III.14.3; the subroutines source code are contained in the same program file element as the driver.

**14.1.2.A INPUT DATA AND DATA BASE:** The SEARCH/ENGAGEREP utility has one input file, the ENGAGEREP. This file is an output of the Concept Evaluation Model (CEM) which is a low resolution theater simulation model. The ENGAGEREP will contain information on not only U.S. forces, but NATO and other non-U.S. forces. Therefore, one of the functions of this utility is to screen out all data which does not pertain to the U.S. forces.

The file is composed of three basic record formats. The first record format identifies that portion of the total force and the particular day being analyzed. The second record format details the total time spent by the Blue forces in one of the 9 combat missions during this specific time period; i.e., 15 days. The third record type specifies the fraction of time spent by U.S. forces during this time period in one of the four combat postures; i.e., Attack, Defend (Defense Intense), Delay or Defense Light (Inactive).

If the ENGAGEREP file is currently active, it will be found on the public mass storage device under the current study's program file using the element name "ENGAGEREP". If the file has been inactivated, it can be found on magnetic tape. The proper tape number and location on the tape can be obtained from the CEM Operator/Analyst.

The record layout is discussed in Volume I. Because of the voluminous quantity of the ENGAGEREP sample data, no example is provided here.

**14.1.2.B OUTPUT DATA AND DATA FILES:** This utility produces one file as output called the SCENARIO/XX. The "XX" portion of this name will be changed to reflect the appropriate CEM run control number as provided by the CEM Operator/Analyst. The SCENARIO/XX file will be used as one of the five input files to the CONTROL/COMPILER utility which follows:

The SCENARIO/XX file will contain information which describes the total time spent by U.S. forces in each of the nine combat missions and the fraction of the total time spent in each of the four combat postures. This information is provided for each of the seven time periods in the study. (It should be noted that the capability does exist to express this information for an additional 3 time periods.)

The SCENARIO/XX file uses 3 record formats to identify and describe each of the seven basic time periods in the study and the distribution of the total time U.S. forces spent within each of the various nine combat missions and the fraction of the total time spent within each of the four combat postures. The file further summarizes at the end of the file, the fraction of time spent in each of the four combat postures for ten time periods.

Figure III.14.4 presents the file layout for the SCENARIO/XX file and examples of the data in the file.

#### **14.1.2.C DATA ELEMENT DICTIONARY:**

The following section identifies and defines all variables used in this utility.

<u>NAME</u>	<u>DEFINITION</u>
LINE	A real variable established as an array in the utility. It is used as a dummy variable into which the first 18 characters from the input record are read and compared to the Constant Code in order to screen out non-US data.
CODE	A Constant dimensioned at 3 members and assigned the following:  CODE (1) = "BLUE" CODE (2) = "PART 1" CODE (3) = "TION 1"
ICYCLE	An integer variable which identifies the day being processed.

<b>ILOOK</b>	An integer variable which is assigned the value of the last day of the current time period being analyzed. It is used to control various breaks and subsequent dispatching within the program.
<b>QREAD</b>	This is the name of a subroutine in the program.
<b>BLUE15</b>	A real variable dimensioned at 9. Each entry in the array will contain the total time, expressed in hours, spent by BLUE forces in this 15 day period in each of the 9 combat situations in which a force can find itself. It should be noted that these 9 combat classifications are simply a finer breakout of the 4 combat postures.
<b>BLUE30 BLUE 90</b>	Same as BLUE15 except they reflect 30 and 90 day periods respectively.
<b>TOTAL</b>	A variable used to accumulate the total amount of time spent in all of the 9 combat situations. This value will be used as the denominator when calculating fractions of total time spent in various situations.
<b>TEMP1</b>	A single dimensioned array of 9 occurrences. This array is used to hold the amount of time spent by US forces in each of the 9 combat situations as they are read in from the ENGAGEREP file.
<b>TEMP</b>	An array of 4 members which is used to hold the fraction of time spent by US forces in each of the 4 combat postures. This data is read from the ENGAGEREP file.
<b>ANSWER</b>	<p>This is an array of 2 dimensions. The first dimension will have 10 occurrences; 1 for each of the maximum of 10 time periods that can be analyzed in the study. The second dimension will identify each of the 4 combat postures that the BLUE forces can assume. Entries in the array will state for each time period the fraction of time spent by forces in each combat posture. This fraction will be a derived value based upon the absolute time spent by forces in each of the 9 combat conditions found in the arrays BLUE15, BLUE-30 or BLUE 90 discussed above. These answer values will be derived following these schemes:</p> <p>ANSWER(X,ATTACK) = The sum of the time spent in the first three combat conditions plus 1/2 of the time spent in the 4th. This summed value is then divided by TOTAL in order to obtain the fraction of time spent.</p>

ANSWER(X,DEFENSE INTENSE) = The sum of the time spent in the 5th and 6th combat conditions plus 1/2 of the time spent in the 4th. This summed value is also divided by TOTAL to derive the fraction of time spent in the second combat posture.

ANSWER(X,DELAY) = The time spent in the 7th combat condition divided by TOTAL.

ANSWER(X,DEFENSE LIGHT) = The sum of the time spent in the 8th and 9th combat conditions divided by TOTAL.

PREV30

A single dimension array with 9 occurrences. Values for this array are st in the subroutine LAST30. These values will be the total time spent by BLUE force, expressed in hours, in each of the 9 combat conditions.

QREAD

A subroutine which is used to manage the reading of the ENGAGEREP file.

LAST30

A subroutine used to store in the array PREV30 the values of the current TEMPI array. This action is required because the utility examines discreet periods of time and the incoming times from the engagerrep file put into TEMPI are accumulated totals. Therefore, by subtracting the values of the PREV30 from the new TEMPI the time spent in the specific 30 day time period can be determined.

J, JK, IJK,  
JKLM

Various integer variables used as subscripts in the utilities.

14.2 OPERATING ENVIRONMENT: This program is implemented on the EXECUTIVE-8 operating system.

14.2.1 SUPPORT SOFTWARE: This utility is written in FORTRAN IV and as such requires the FORTRAN IV compiler and the facilities of the UNIVAC 1100 system.

14.2.2 I/O DEVICES: This utility uses as input a file which may or may not reside on disk. If the file is active it will reside on disk. If the file is inactive it will reside on tape. The significance of this situation, beyond the fact that differing input device will be required, is that different runstreams are used to execute the program. Thus it is anticipated that users may execute the wrong runstream and the system will either not be able to find the designated input file or an out dated input file may be used in the utility and incorrect results may be written to the output file. The output file will be written on to disk.

14.3 MAINTENANCE PROCEDURES: The program is maintained on the system by the MPP analyst. There are no special maintenance procedures.

14.3.1 PROGRAMMING CONVENTIONS: This is one of the few utilities in the MPP which uses subroutines in an effort to isolate repeated procedures. The utility has two of these routines. One, "QREAD," isolates the majority of the reads of the input file \*\*ENGAGEREP. The second routine "LAST30" is used to "age" or move to temporary holding array values of the current period which will be used in subsequent calculations.

The utility is written in FORTRAN IV and follows FORTRAN programming conventions.

14.3.2 INTERNAL ERROR ROUTINES: There are no explicit error handling routines written into this utility. Errors detected during program execution will be identified and the appropriate system messages will be printed in the PRINT\$ file.

# SEARCH/ENGAGEREP STRUCTURE

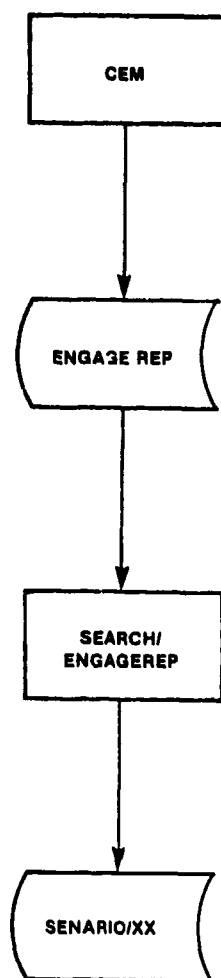


Figure III.14.1

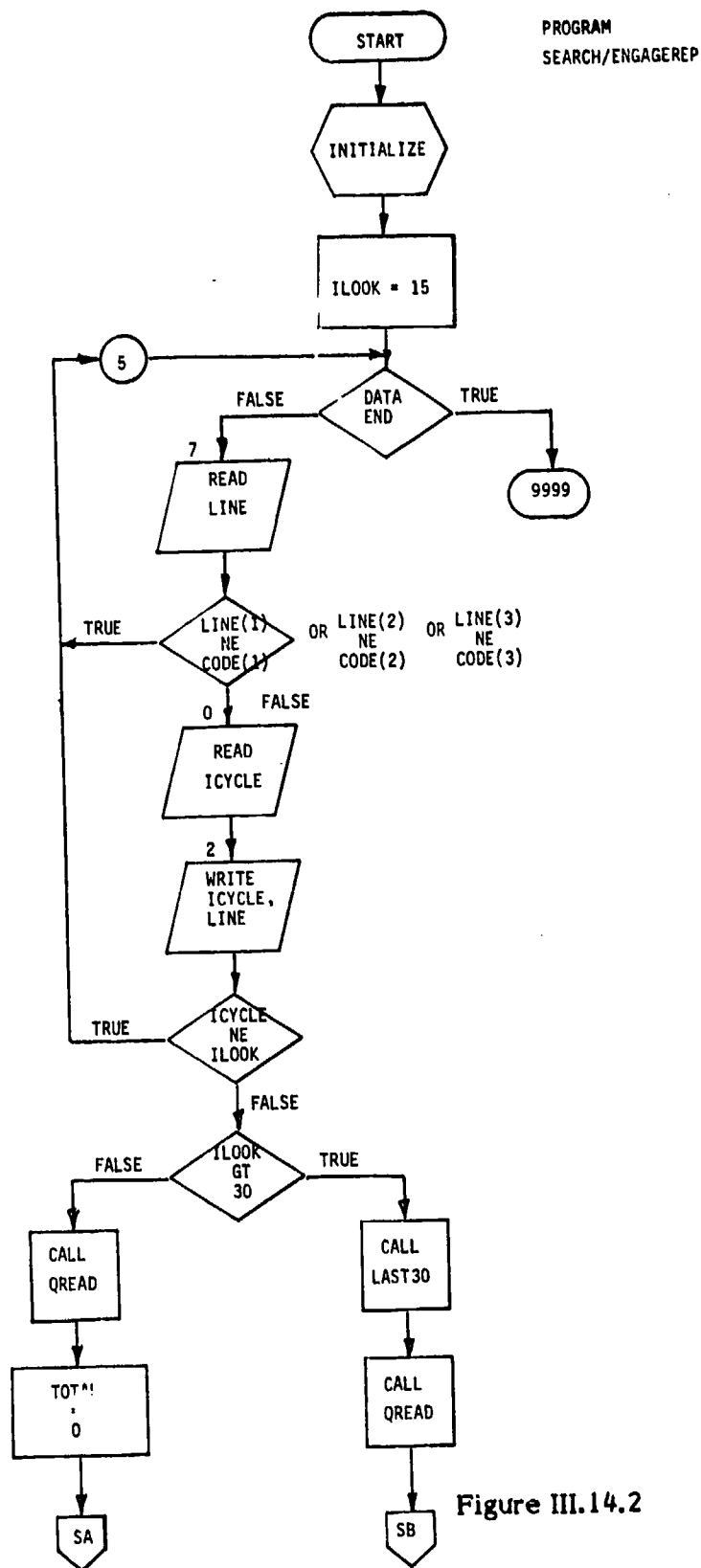


Figure III.14.2

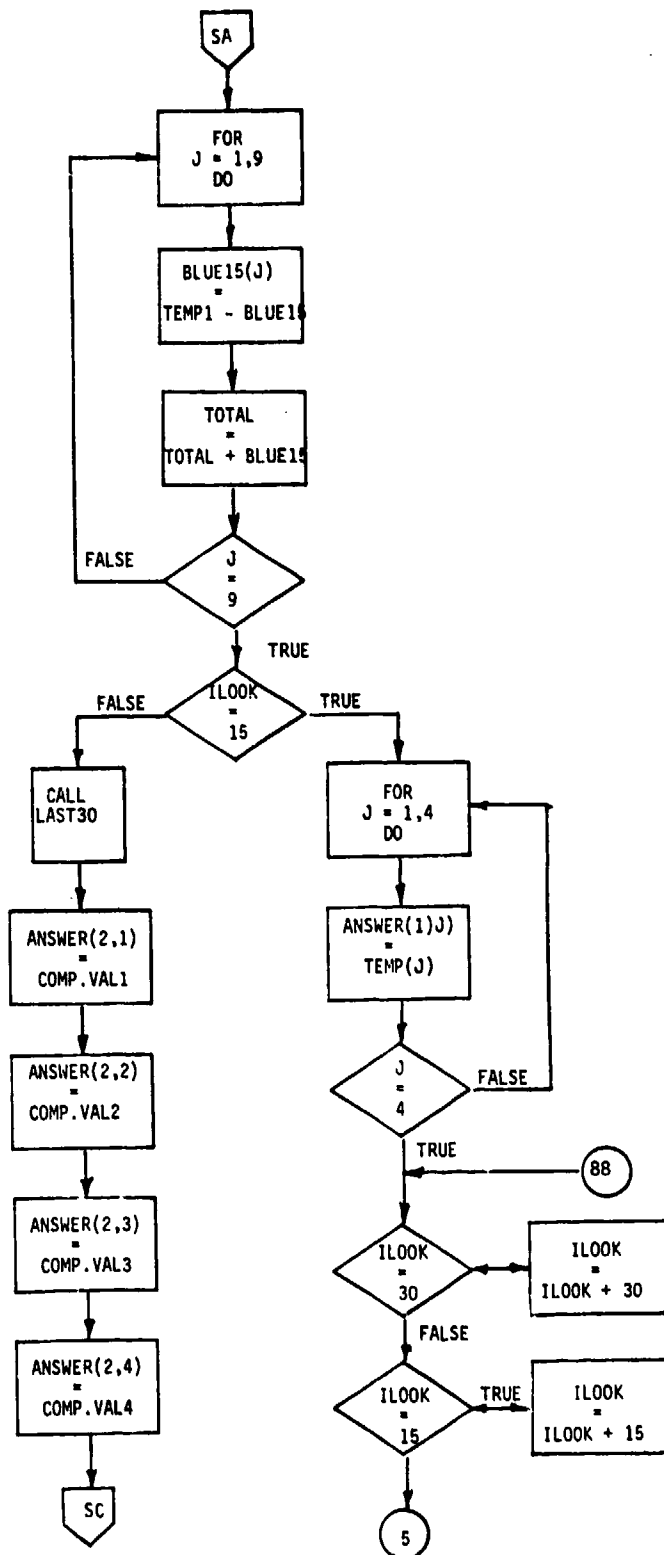


Figure III.14.2 (Cont)

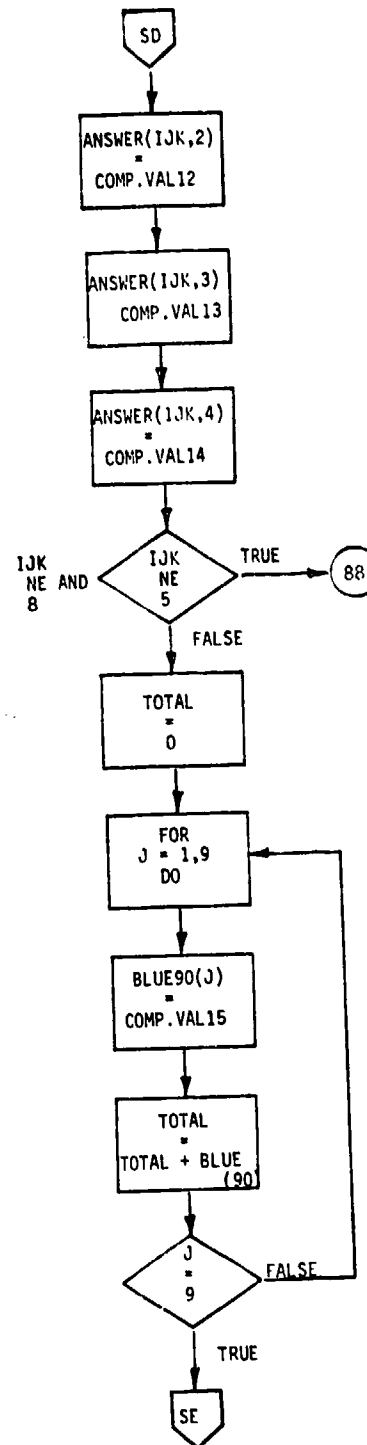
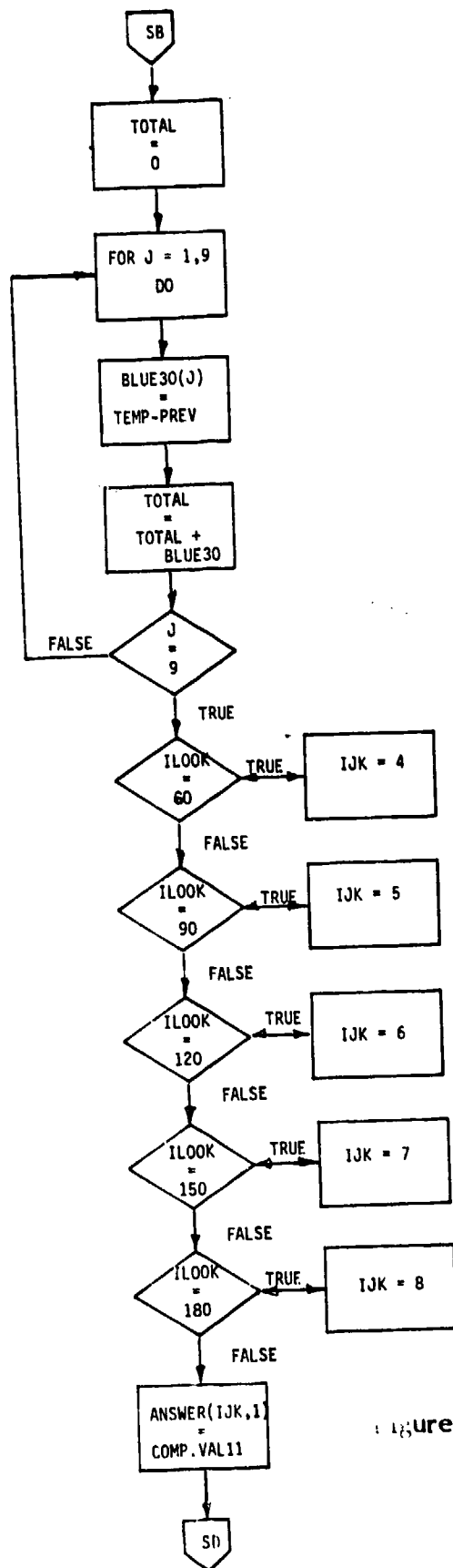


Figure III.14.2 (Cont)

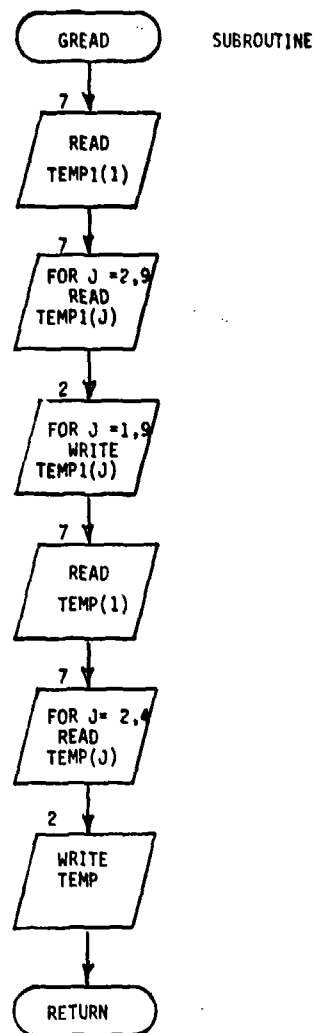
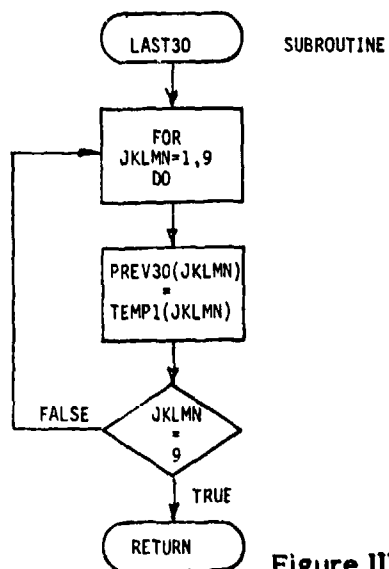
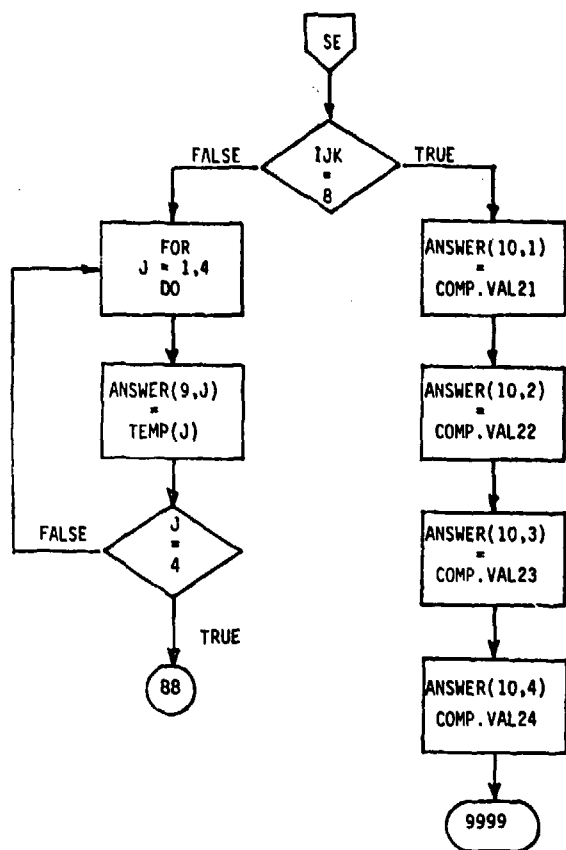


Figure III.14.2 (Cont)

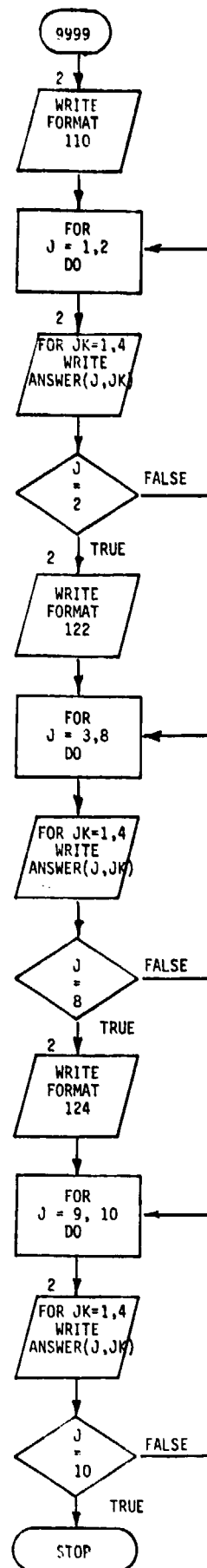
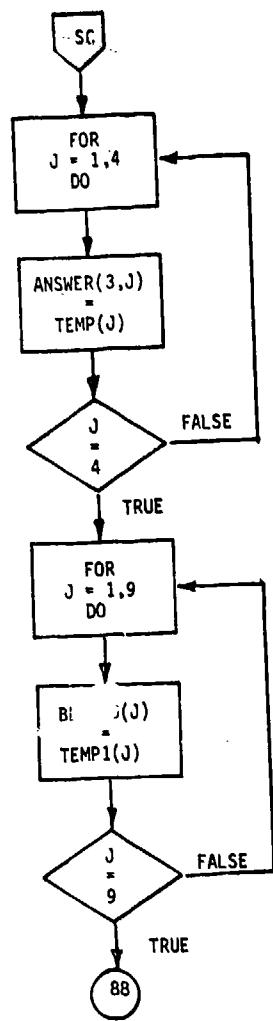


Figure III.14.2 (Cont)

UNCLASSIFIED\*\*\*FILE NAME:62XGT ELEMENT NAME:SEARCH/ENGAGEREP\*\*\*UNCLASSIFIED

```

1:C*****
2:C*****
3:C*****THIS PROGRAM SEARCHES THROUGH THE ENGAGEMENT REPORT FROM CEM.
4:C*****AND COMPUTES FOR EACH OF THE SEVEN TIME PERIODS OF THE STUDY
5:C*****THE FRACTION OF TIME SPENT BY THE TOTAL US FORCE IN EACH
6:C*****OF THE FOUR COMBAT POSTURES (ATTACK, DEFENSE INTENSE, DELAY
7:C*****AND DEFENSE LIGHT).
8:C*****THE OUTPUT OF THIS UTILITY (US FORCE SCENARIO) WILL BE
9:C*****USED AS INPUT BY THE UTILITY "62XGT.CONTROL/COMPILE".
10:C*****NOTE THAT THERE ARE TWO DISTINCTLY DIFFERENT RUN STREAMS
11:C*****FOR THIS UTILITY, USE DEPENDENT ON THE CONDITION OF THE CEM
12:C*****FILE "ENGAGEREP" WITHIN THE COMPUTER'S FILING SYSTEM.
13:C*****ACTIVE (ON PUBLIC PACK) OR INACTIVE (STORED ON TAPE).
14:C*****
15:C*****
16:      DIMENSION LINE(3),ANSWER(10,4),TEMP(4),TEMP1(9),CODE(3),
17:      BLUE15(9),BLUE30(9),BLUE90(9),PREV30(9)
18:      REAL LINE
19:      DATA CODE/6H BLUE,6H PARTI,6H TION 1/
20:      ILOOK=15
21:5      READ(7,10,END=9999)LINE
22:10      FORMAT(3X,3A6)
23:      IF((LINE(1).NE.CODE(1)).OR.(LINE(2).NE.CODE(2)).OR.(LINE(3).NE.
24:      CODE(3)))GO TO 5
25:      READ(6,11)ICYCLE
26:11      FORMAT(7UX,I4)
27:      WRITE(2,14)ICYCLE,LINE
28:14      FORMAT(1X,'ICYCLE=',I4,10X,3A6)
29:      IF(ICYCLE.NE.ILOOK)GO TO 5
30:      IF(ILOOK.GT.30)GO TO 40
31:C*****
32:C*****TAKE CARE OF 15 DAY CYCLES
33:C*****
34:      CALL GREAD
35:      TOTAL=0
36:      DO 20 J=1,9
37:      BLUE15(J)=TEMP1(J)-BLUE15(J)
38:      TOTAL=TOTAL+BLUE15(J)
39:20      CONTINUE
40:      IF(ILOOK.EQ.15)GO TO 30
41:C*****
42:C*****CYCLE=30
43:C*****
44:      CALL LAST30
45:      ANSWER(2,1)=(BLUE15(1)+BLUE15(2)+BLUE15(3)+0.5*BLUE15(4))/TOTAL
46:      ANSWER(2,2)=(BLUE15(5)+BLUE15(6)+0.5*BLUE15(4))/TOTAL
47:      ANSWER(2,3)=BLUE15(7)/TOTAL
48:      ANSWER(2,4)=(BLUE15(8)+BLUE15(9))/TOTAL
49:C*****
50:C*****ABOVE IS ANSWERS FOR 2ND 15 DAYS,NEXT->ANSWERS FOR 1ST 30-DAYS.
51:C*****
52:      DO 29 J=1,4
53:      ANSWER(3,J)=TEMP(J)
54:29      CONTINUE
55:      DO 23 J=1,9
56:      BLUE30(J)=TEMP1(J)
57:23      CONTINUE

```

Figure III.14.3

UNCLASSIFIED\*\*\*FILE NAME:82XGT ELEMENT NAME:SEARCH/ENGADREP\*\*\*UNCLASSIFIED

```

58:      GO TO 88
59:C*****
60:C*****ICYCLE=15
61:C*****COMPUTE ANSWERS FOR 1ST 15-DAYS
62:C*****
63:30    DO 32 J=1,4
64:      ANSWER(1,J)=TEMP(J)
65:32    CONTINUE
66:      GO TO 89
67:C*****
68:C*****ICYCLE>30
69:C*****
70:40    CALL LAST30
71:      CALL GREAD
72:      TOTAL=0
73:      DO 41 J=1,9
74:        BLUE30(J)=TEMP(J)-PREV 30(J)
75:        TOTAL=TOTAL+BLUE30(J)
76:41    CONTINUE
77:      IF(LOOK.EQ.60)IJK=4
78:      IF(LOOK.EQ.90)IJK=5
79:      IF(LOOK.EQ.120)IJK=6
80:      IF(LOOK.EQ.150)IJK=7
81:      IF(LOOK.EQ.180)IJK=8
82:      ANSWER(IJK,1)=(BLUE30(1)+BLUE30(2)+BLUE30(3)+0.5*BLUE30(4))/TOTAL
83:      ANSWER(IJK,2)=(BLUE30(5)+BLUE30(6)+0.5*BLUE30(4))/TOTAL
84:      ANSWER(IJK,3)=BLUE30(7)/TOTAL
85:      ANSWER(IJK,4)=(BLUE30(8)+BLUE30(9))/TOTAL
86:      IF(IJK.NE.5).AND.(IJK.NE.8)GO TO 88
87:C*****
88:C*****CYCLE IS ALSO A MULTIPLE OF 90 (90 OR 180)
89:C*****
90:      TOTAL=0
91:      DO 45 J=1,9
92:        BLUE90(J)=TEMP(J)-BLUE 90(J)
93:        TOTAL=TOTAL+BLUE90(J)
94:45    CONTINUE
95:      IF(IJK.EQ.8)GO TO 77
96:C*****
97:C*****ICYCLE=90
98:C*****
99:      DO 47 J=1,4
100:      ANSWER(9,J)=TEMP(J)
101:47    CONTINUE
102:      GO TO 88
103:C*****
104:C*****ICYCLE=180
105:C*****
106:77    ANSWER(10,1)=(BLUE90(1)+BLUE90(2)+BLUE90(3)+0.5*BLUE90(4))/TOTAL
107:      ANSWER(10,2)=(BLUE90(5)+BLUE90(6)+0.5*BLUE90(4))/TOTAL
108:      ANSWER(10,3)=BLUE90(7)/TOTAL
109:      ANSWER(10,4)=(BLUE90(8)+BLUE90(9))/TOTAL
110:      GO TO 9999
111:C*****
112:C*****
113:88    IF(LOOK.GE.30)LOOK=LOOK+30
114:      IF(LOOK.EQ.15)LOOK=LOOK+15

```

Figure III.14.3 (Cont)

UNCLASSIFIED\*\*\*FILE NAME:182XGT ELEMENT NAME:SEARCH/ENGACREP\*\*\*UNCLASSIFIED

```

115:      GO TO 5
116:C*****
117:C*****
118:C*****PRINT OUT ANSWERS
119:C*****
120:C*****
121:1999  WRITE (2,110)
122:110   FORMAT(1H1,4X,'ATTACK',4X,'DEFEND',5X,'DELAY',2X,'INACTIVE' /
123:      210X,'15 DAYS')
124:      DO 190 J=1,2
125:      WRITE (2,120)(ANSWER(J,JK),JK=1,4)
126:120   FORMAT(1X,4F10.4)
127:190   CONTINUE
128:C*****
129:C*****
130:      WRITE (2,122)
131:122   FORMAT(////1X,'30 DAYS')
132:      DO 192 J=3,8
133:      WRITE (2,120)(ANSWER(J,JK),JK=1,4)
134:192   CONTINUE
135:C*****
136:C*****
137:      WRITE (2,124)
138:124   FORMAT(////1X,'90 DAYS')
139:      DO 194 J=9,10
140:      WRITE (2,120)(ANSWER(J,JK),JK=1,4)
141:194   CONTINUE
142:      STOP

```

```

143:C*****
144:C*****
145:C*****SUBROUTINES
146:C*****
147:C*****
148:      SUBROUTINE GREAD
149:      READ(7,26)TEMP1(1)
150:26     FORMAT(/////////69X,F19.1)
151:27     FORMAT(25X,F7.1)
152:      READ(7,24)(TEMP1(J),J=2,9)
153:24     FORMAT(/69X,F18.1)
154:      WRITE (2,27)(TEMP1(J),J=1,9)
155:      READ(7,25)TEMP(1)
156:25     FORMAT(/////////95X,F19.3)
157:      READ(7,21)(TEMP(J),J=2,4)
158:21     FORMAT(95X,F19.3)
159:      WRITE (2,889)TEMP
160:889    FORMAT(40X,F8.3)
161:      RETURN
162:      SUBROUTINE LAST30
163:      DO 370 JKLMN=1,9
164:      PREV30(JKLMN)=TEMP1(JKLMN)
165:370    CONTINUE
166:      RETURN
167:      END

```

Figure III.14.3 (Cont)

UNCLASSIFIED\*\*\*EXAMPLE OF THE OUTPUT FROM UTILITY SEARCH/ENCASOFT\*\*\*UNCLASSIFIED

1:	ICVCLF=	1	BLUE PARTITION 1	
2:	ICVCLF=	2	BLUE PARTITION 1	
3:	ICVCLF=	3	BLUE PARTITION 1	
4:	ICVCLF=	4	BLUE PARTITION 1	
5:	ICVCLF=	5	BLUE PARTITION 1	
6:	ICVCLF=	6	BLUE PARTITION 1	
7:	ICVCLF=	7	BLUE PARTITION 1	
8:	ICVCLF=	8	BLUE PARTITION 1	
9:	ICVCLF=	9	BLUE PARTITION 1	
10:	ICVCLF=	10	BLUE PARTITION 1	
11:	ICVCLF=	11	BLUE PARTITION 1	
12:	ICVCLF=	12	BLUE PARTITION 1	
13:	ICVCLF=	13	BLUE PARTITION 1	
14:	ICVCLF=	14	BLUE PARTITION 1	
15:	ICVCLF=	15	BLUE PARTITION 1	
16:			.0	
17:			.0	
18:			.0	
19:			.0	
20:			243.7	
21:			339.2	
22:			39.6	
23:			2570.6	
24:			207.0	
25:				.000
26:				.171
27:				.012
28:				.817
29:	ICVCLF=	16	BLUE PARTITION 1	
30:	ICVCLF=	17	BLUE PARTITION 1	
31:	ICVCLF=	18	BLUE PARTITION 1	
32:	ICVCLF=	19	BLUE PARTITION 1	
33:	ICVCLF=	20	BLUE PARTITION 1	
34:	ICVCLF=	21	BLUE PARTITION 1	
35:	ICVCLF=	22	BLUE PARTITION 1	
36:	ICVCLF=	23	BLUE PARTITION 1	
37:	ICVCLF=	24	BLUE PARTITION 1	
38:	ICVCLF=	25	BLUE PARTITION 1	
39:	ICVCLF=	26	BLUE PARTITION 1	
40:	ICVCLF=	27	BLUE PARTITION 1	
41:	ICVCLF=	28	BLUE PARTITION 1	
42:	ICVCLF=	29	BLUE PARTITION 1	
43:	ICVCLF=	30	BLUE PARTITION 1	
44:			213.9	
45:			339.5	
46:			266.3	
47:			30.3	
48:			475.9	
49:			644.0	
50:			70.7	
51:			5714.4	
52:			457.0	
53:				.109
54:				.145
55:				.009
56:				.737
57:	ICVCLF=	31	BLUE PARTITION 1	

Figure III.14.4

UNCLASSIFIED\*\*\*EXAMPLE OF THE OUTPUT FROM UTILITY SEARCH/ENGAGE\*\*\*UNCLASSIFIED

540: ICVCLE = 32	BLUE PARTITION 1
541: ICVCLE = 33	BLUE PARTITION 1
542: ICVCLE = 34	BLUE PARTITION 1
543: ICVCLE = 35	BLUE PARTITION 1
544: ICVCLE = 36	BLUE PARTITION 1
545: ICVCLE = 37	BLUE PARTITION 1
546: ICVCLE = 38	BLUE PARTITION 1
547: ICVCLE = 39	BLUE PARTITION 1
548: ICVCLE = 40	BLUE PARTITION 1
549: ICVCLE = 41	BLUE PARTITION 1
550: ICVCLE = 42	BLUE PARTITION 1
551: ICVCLE = 43	BLUE PARTITION 1
552: ICVCLE = 44	BLUE PARTITION 1
553: ICVCLE = 45	BLUE PARTITION 1
554: ICVCLE = 46	BLUE PARTITION 1
555: ICVCLE = 47	BLUE PARTITION 1
556: ICVCLE = 48	BLUE PARTITION 1
557: ICVCLE = 49	BLUE PARTITION 1
558: ICVCLE = 50	BLUE PARTITION 1
559: ICVCLE = 51	BLUE PARTITION 1
560: ICVCLE = 52	BLUE PARTITION 1
561: ICVCLE = 53	BLUE PARTITION 1
562: ICVCLE = 54	BLUE PARTITION 1
563: ICVCLE = 55	BLUE PARTITION 1
564: ICVCLE = 56	BLUE PARTITION 1
565: ICVCLE = 57	BLUE PARTITION 1
566: ICVCLE = 58	BLUE PARTITION 1
567: ICVCLE = 59	BLUE PARTITION 1
568: ICVCLE = 60	BLUE PARTITION 1
569:	650.5
570:	1418.0
571:	1110.5
572:	132.1
573:	1055.5
574:	1623.6
575:	114.2
576:	14913.6
577:	1270.0
578:	
579:	.146
580:	.123
581:	.007
582:	.724
583:	
1000: ICVCLE = 61	BLUE PARTITION 1
1001: ICVCLE = 62	BLUE PARTITION 1
1002: ICVCLE = 63	BLUE PARTITION 1
1003: ICVCLE = 64	BLUE PARTITION 1
1004: ICVCLE = 65	BLUE PARTITION 1
1005: ICVCLE = 66	BLUE PARTITION 1
1006: ICVCLE = 67	BLUE PARTITION 1
1007: ICVCLE = 68	BLUE PARTITION 1
1008: ICVCLE = 69	BLUE PARTITION 1
1009: ICVCLE = 70	BLUE PARTITION 1
1010: ICVCLE = 71	BLUE PARTITION 1
1011: ICVCLE = 72	BLUE PARTITION 1
1012: ICVCLE = 73	BLUE PARTITION 1
1013: ICVCLE = 74	BLUE PARTITION 1
1014: ICVCLE = 75	BLUE PARTITION 1

Figure III.14.4 (Cont)

UNCLASSIFIED\*\*\*EXAMPLE OF THE OUTPUT FROM UTILITY SEARCH/ENGAGEMENT\*\*\*UNCLASSIFIED

115: ICYCLE = 76	BLUE PARTITION 1
116: ICYCLE = 77	BLUE PARTITION 1
117: ICYCLE = 78	BLUE PARTITION 1
118: ICYCLE = 79	BLUE PARTITION 1
119: ICYCLE = 80	BLUE PARTITION 1
120: ICYCLE = 81	BLUE PARTITION 1
121: ICYCLE = 82	BLUE PARTITION 1
122: ICYCLE = 83	BLUE PARTITION 1
123: ICYCLE = 84	BLUE PARTITION 1
124: ICYCLE = 85	BLUE PARTITION 1
125: ICYCLE = 86	BLUE PARTITION 1
126: ICYCLE = 87	BLUE PARTITION 1
127: ICYCLE = 88	BLUE PARTITION 1
128: ICYCLE = 89	BLUE PARTITION 1
129: ICYCLE = 90	BLUE PARTITION 1
130:	1146.8
131:	2887.2
132:	2846.2
133:	207.8
134:	1107.0
135:	2427.7
136:	200.6
137:	28258.7
138:	2066.0
139:	
140:	.153
141:	.093
142:	.006
143:	.749
143: ICYCLE = 91	BLUE PARTITION 1
144: ICYCLE = 92	BLUE PARTITION 1
145: ICYCLE = 93	BLUE PARTITION 1
146: ICYCLE = 94	BLUE PARTITION 1
147: ICYCLE = 95	BLUE PARTITION 1
148: ICYCLE = 96	BLUE PARTITION 1
149: ICYCLE = 97	BLUE PARTITION 1
150: ICYCLE = 98	BLUE PARTITION 1
151: ICYCLE = 99	BLUE PARTITION 1
152: ICYCLE = 100	BLUE PARTITION 1
153: ICYCLE = 101	BLUE PARTITION 1
154: ICYCLE = 102	BLUE PARTITION 1
155: ICYCLE = 103	BLUE PARTITION 1
156: ICYCLE = 104	BLUE PARTITION 1
157: ICYCLE = 105	BLUE PARTITION 1
158: ICYCLE = 106	BLUE PARTITION 1
159: ICYCLE = 107	BLUE PARTITION 1
160: ICYCLE = 108	BLUE PARTITION 1
161: ICYCLE = 109	BLUE PARTITION 1
162: ICYCLE = 110	BLUE PARTITION 1
163: ICYCLE = 111	BLUE PARTITION 1
164: ICYCLE = 112	BLUE PARTITION 1
165: ICYCLE = 113	BLUE PARTITION 1
166: ICYCLE = 114	BLUE PARTITION 1
167: ICYCLE = 115	BLUE PARTITION 1
168: ICYCLE = 116	BLUE PARTITION 1
169: ICYCLE = 117	BLUE PARTITION 1
170: ICYCLE = 118	BLUE PARTITION 1
171: ICYCLE = 119	BLUE PARTITION 1

Figure III.14.4 (Cont)

UNCLASSIFIED\*\*\*EXAMPLE OF THE OUTPUT FROM UTILITY SEARCH/ENGAGEMENT\*\*\*UNCLASSIFIED

172: ICYCLE= 120	BLUE PARTITION 1	
173:	1300.8	
174:	4022.7	
175:	2511.2	
176:	212.4	
177:	1325.6	
178:	3353.4	
179:	572.5	
180:	42712.6	
181:	2773.0	
182:		.135
183:		.081
184:		.010
185:		.774
186: ICYCLE= 121	BLUE PARTITION 1	
187: ICYCLE= 122	BLUE PARTITION 1	
188: ICYCLE= 123	BLUE PARTITION 1	
189: ICYCLE= 124	BLUE PARTITION 1	
190: ICYCLE= 125	BLUE PARTITION 1	
191: ICYCLE= 126	BLUE PARTITION 1	
192: ICYCLE= 127	BLUE PARTITION 1	
193: ICYCLE= 128	BLUE PARTITION 1	
194: ICYCLE= 129	BLUE PARTITION 1	
195: ICYCLE= 130	BLUE PARTITION 1	
196: ICYCLE= 131	BLUE PARTITION 1	
197: ICYCLE= 132	BLUE PARTITION 1	
198: ICYCLE= 133	BLUE PARTITION 1	
199: ICYCLE= 134	BLUE PARTITION 1	
200: ICYCLE= 135	BLUE PARTITION 1	
201: ICYCLE= 136	BLUE PARTITION 1	
202: ICYCLE= 137	BLUE PARTITION 1	
203: ICYCLE= 138	BLUE PARTITION 1	
204: ICYCLE= 139	BLUE PARTITION 1	
205: ICYCLE= 140	BLUE PARTITION 1	
206: ICYCLE= 141	BLUE PARTITION 1	
207: ICYCLE= 142	BLUE PARTITION 1	
208: ICYCLE= 143	BLUE PARTITION 1	
209: ICYCLE= 144	BLUE PARTITION 1	
210: ICYCLE= 145	BLUE PARTITION 1	
211: ICYCLE= 146	BLUE PARTITION 1	
212: ICYCLE= 147	BLUE PARTITION 1	
213: ICYCLE= 148	BLUE PARTITION 1	
214: ICYCLE= 149	BLUE PARTITION 1	
215: ICYCLE= 150	BLUE PARTITION 1	
216:	2183.8	
217:	6360.7	
218:	4459.3	
219:	227.2	
220:	1327.0	
221:	3555.6	
222:	582.3	
223:	54420.1	
224:	3879.0	
225:		.171
226:		.065
227:		.008
228:		.757

Figure III.14.4 (Cont)

UNCLASSIFIED\*\*\*EXAMPLE OF THE OUTPUT FROM UTILITY SEARCH/ENCIPHER\*\*\*UNCLASSIFIED

229:	ICYCLE = 151	BLUE PARTITION 1
230:	ICYCLE = 152	BLUE PARTITION 1
231:	ICYCLE = 153	BLUE PARTITION 1
232:	ICYCLE = 154	BLUE PARTITION 1
233:	ICYCLE = 155	BLUE PARTITION 1
234:	ICYCLE = 156	BLUE PARTITION 1
235:	ICYCLE = 157	BLUE PARTITION 1
236:	ICYCLE = 158	BLUE PARTITION 1
237:	ICYCLE = 159	BLUE PARTITION 1
238:	ICYCLE = 160	BLUE PARTITION 1
239:	ICYCLE = 161	BLUE PARTITION 1
240:	ICYCLE = 162	BLUE PARTITION 1
241:	ICYCLE = 163	BLUE PARTITION 1
242:	ICYCLE = 164	BLUE PARTITION 1
243:	ICYCLE = 165	BLUE PARTITION 1
244:	ICYCLE = 166	BLUE PARTITION 1
245:	ICYCLE = 167	BLUE PARTITION 1
246:	ICYCLE = 168	BLUE PARTITION 1
247:	ICYCLE = 169	BLUE PARTITION 1
248:	ICYCLE = 170	BLUE PARTITION 1
249:	ICYCLE = 171	BLUE PARTITION 1
250:	ICYCLE = 172	BLUE PARTITION 1
251:	ICYCLE = 173	BLUE PARTITION 1
252:	ICYCLE = 174	BLUE PARTITION 1
253:	ICYCLE = 175	BLUE PARTITION 1
254:	ICYCLE = 176	BLUE PARTITION 1
255:	ICYCLE = 177	BLUE PARTITION 1
256:	ICYCLE = 178	BLUE PARTITION 1
257:	ICYCLE = 179	BLUE PARTITION 1
258:	ICYCLE = 180	BLUE PARTITION 1
259:		2570.0
260:		7537.2
261:		4716.5
262:		233.2
263:		1338.9
264:		3641.9
265:		529.5
266:		70023.7
267:		4503.0
268:		.157
269:		.054
270:		.006
271:		.783
272:	ATTACK DEFEND DELAY INACTIVE	
273:	15 DAYS	
274:	.0000 .1810 .0320 .7870	
275:	.2926 .0246 .1070 .5757	
276:		
277:		
278:		
279:		
280:	30 DAYS	
281:	.2090 .2450 .1090 .4370	
282:	.2665 .2708 .0004 .5163	
283:	.2600 .1563 .0004 .5502	
284:	.1963 .1557 .1190 .5290	
285:	.3649 .1121 .0105 .4924	
286:	.2016 .1050 .1004 .4930	
287:		
288:		
289:		
290:		
291:	90 DAYS	
292:	.2530 .1930 .1060 .4490	
293:	.2609 .1243 .1066 .4092	

Figure III.14.4 (Cont)

THIS PAGE IS INTENTIONALLY LEFT BLANK

## CHAPTER 15

### Utility - CEM/DATA

**15.1 DESCRIPTION OF PROCESSING:** This program performs math functions on the input data.

**15.1.1 PURPOSE/FUNCTIONS:** The purpose of this utility is to produce the CEM/DATA file which details, time period by time period, equipment authorization levels and loss rates for each specific item equipment type within the four major classifications of combat equipment, i.e., tanks, armored personnel carriers (APC), helicopters, and anti-tank/mortar weapons (ATM). This data is, in turn, used as input to the utility CEM/LOSS.

This utility and the following CEM/LOSS utility, are unique within the Materiel Postprocessor in that they share and are executed from the same runstream, entitled CEM/LOSSES, which is cataloged as an element under the program file CSTART\*82XQT. The output file from the CEM/DATA is used immediately as input to the CEM/LOSSES and is released immediately. Only the output from CEM/LOSSES is permanently cataloged. Therefore, if processing is interrupted at any point in the running of this series of utilities, both must be rerun. Refer to Volume I for the program runstream.

**15.1.2 PROGRAM INPUT/OUTPUT STRUCTURE:** The overall structure of the utility is pictured in FIGURE III.15.1. The logic flow followed by the utility is depicted in FIGURE III.15.2. The source code of the utility is found in FIGURE III.15.3.

**15.1.2.A INPUT DATA AND DATA BASE:** There is only one input file for this utility. It is the CEM/DATA report file which is produced exclusively for the WARF from the CEM LOG REPORT (LOGREP). The file should be cataloged as an element under the current study's program file, in this present case SECRET\*82WARF88, using the element name CEM/DATA.

The file details the number of items of equipment authorized and the CEM loss rates for up to 12 individual models of four major items of combat equipment. The four major items of combat equipment are tanks, armored personnel carriers (APC), helicopters and antitank & mortar weapons (ATM). An example of one of the 12 individual models of one of these major combat items would be the M1 Model Tank.

Figure III.15.4 details the file layout for the CEM Data Record file.

**15.1.2.B OUTPUT DATA AND DATA FILES:** This utility produces one output file, 83CEMLOSS1. This file details for each of the equipment model types (a maximum of 12) within the four major Combat equipment categories, (i.e., tanks, APC's helicopters and ATM's) the average number of items authorized for a specific time period and the percentage loss rate for that period.

As can be seen, the utility has taken the input order of battle data and accumulated into several time periods of the battle. These time periods are further classified into the Intensive Period and Sustaining Period of the battle. The Intensive Period is the first 90 days and the sustaining Period is the last 90 days.

The file, pictured in Figure III.15.5 is highly formatted in order that it can be read as a report for data verification before it is used as an input file to the following utility CEM/LOSSES. As a result, the file is self explanatory.

The file is unique in that it is the only file within the system that is not permanently saved; it is read immediately by the following utility CEM/LOSSES. Thus if processing is interrupted in either of these two utilities both will have to be rerun.

#### 15.4.2.C DATA ELEMENT DICTIONARY:

The following section identifies and defines all variables used in this utility.

TK	A single dimension variable of 12 occurrences. Each occurrence specifies the daily authorized quantity of a specific model of tank. These values are read from the CEM LOG REPORT. The unit of measure used is TANKS.
TKL	A single dimension array of 12 occurrences. Each occurrence will denote the daily 12 hour losses of each of the 12 models of tanks in the study. These values are read from the CEM LOG REPORT. The unit of measure is TANKS.
TTK	A single dimension array of 12 occurrences. Each occurrence specifies the cumulative total of authorized tanks for a 30 day period. It equals the summation of the daily authorizations.
TKCUM	<p>A 2 dimension array. The first dimension contains 12 occurrences, 1 occurrence for each model of tank that can be played. The second dimension identifies each of the 6 30-day blocks of time that can be played. Entries in the array accumulate the authorized quantity data for the 12 tank models. This cumulative total is equal to 1/2 of 60 Division Cycles (TTK). Therefore:</p> $TK\ CUM(MODEL, 30\ DAY\ BLOCK) = TTK(MODEL)/2.$
TTKL	A single dimension array of 12 occurrences which is used to accumulate tank losses by model. Unit of measure is the TANK.
TKLCUM	A 2 dimension array which is used to accumulate loss data for tanks by model and 30 day block, similar to TTKCUM discussed above. The cumulative total is

equal to 1/2 of 60 Division Cycles (TTKL). Thus:

$TKLCUM(MODEL, 30 \text{ DAY BLOCK}) = TTKL(MODEL) / 2.$

- AP** A 1 dimension array of 12 occurrences. Each occurrence specifies the daily authorized quantities of up to 12 specific models of Armored Personnel Carriers. This data is read from the CEM LOG REPORT. Unit of Measure is APC.
- APL** A single dimension array of 12 occurrences. Each occurrence specifies for each of the up to 12 models of APC losses sustained. This data is read from the CEM LOG REPORT. Unit of measure is APC. **TAPA** single dimension array of 12 occurrences. Each occurrence denotes the accumulated number of APC for each of the 12 models over 30 days. This is a derived value which results from the sum of authorized AP quantities for specific APC models. Unit of measure is APC.
- TAPL** A single dimension array of 12 occurrences. Each occurrence denotes the accumulated losses of specific models of APC in the study. This is a derived value resulting from the sum of APL (APC losses) for specific models. Unit of measure is APC.
- APCUM** A 2 dimension array. Each entry in the array specifies the cumulative number of items broken out by APC model and 30 day block. Entries in the array accumulate the authorized quantities of APC by model and 30 day block. The values are derived using a formula:
- $APCUM(MODEL, 30 \text{ DAY BLOCK}) = TAP(MODEL) / 2.$
- The unit of measure is APC.
- APLCOM** A 2 dimension array. Each entry in the array specifies, for each APC model type and 30 day block, the accumulated losses. This value is derived by a simple assignment of the current TAPL (MODEL) to the APLCUM (MODEL, 30 DAY BLOCK). The unit of measure is APC.
- HE** This is a single dimension array with 5 occurrences. Entries in the array denote the authorized quantities for each of 5 Helicopter models in the study. This data is obtained from the CEM LOG REPORT file. The unit of measure is HELICOPTERS.
- HEL** This is a single dimension array with 5 occurrences. Entries in the array specify losses sustained for each

of the 5 Helicopter models in the study. This data is read from the CEM LOG REPORT file. Unit of measure is HELICOPTERS.

THE

This is a single dimension array with 5 occurrences. Each entry in the array denotes the cumulative number of helicopters by model for 30 days. This is a derived value resulting from the summation of helicopter quantity authorization data for the current 30 day period. The unit of measure is HELICOPTER.

THEL

This is a single dimension array of 5 occurrences. Each entry specifies losses for each model of Helicopter. This data read from the CEM LOSSES REPORT file. Unit of measure is the HELICOPTER.

HECUM

This is a 2 dimension array, which specifies for each of the 5 Helicopter models and 30 Day Block of time the total quantity authorized. This is a derived value. Entries in the array are equal to 1/2 of the 60 division Cycles (THE) using the equation:

$$\text{HECUM}(\text{MODEL}, 30 \text{ DAY BLOCK}) = \text{THE}(\text{MODEL})/2$$

The unit of measure is HELICOPTER.

HELCUM

This is a 2 dimension array which specifies losses for each Helicopter model and 30 Day Block of time. This value is assigned by the utility using the current Helicopter model losses, (THEL). The unit of measure is HELICOPTER.

AT

This is a single dimension array of 12 occurrences. Each entry denotes for each of the 12 possible models of Anti-tank and Mortar (AT) Weapon System the authorized quantity. This value will be read from the CEM LOG REPORT input file. The unit of measure is ATM.

ATL

A single dimension array of 12 occurrences. Each entry in the array indicates the number of losses that were encountered for each ATM model in the current 30 day period. This value is read from the CEM LOG REPORT input file. The unit of measure is ATM.

TAT

This is a single dimension array of 12 occurrences. Each occurrence specifies for each ATM model the total quantity for a 30 day period. The value is derived from summing the ATM's authorized for the current 30 day period. The unit of measure is ATM.

TATL	A single dimension array of 12 occurrences. Each occurrence indicates for each of the 12 ATM models the total losses sustained during the current 30 day period. This is a value derived by summing the individual ATL values. The unit of measure is ATM.
ATCUM	This is a 2 dimension array which specifies for each ATM model and 30 Day Block of time the accumulated quantity authorized. The values are derived by dividing the 60 Division Cycle by 2. Thus:  $\text{ATCUM}(\text{MODEL}, 30 \text{ DAY BLOCK}) = \text{TAT}(\text{MODEL})/2$ <p>Unit of measure is ATM.</p>
ATLCUM	This is a 2 dimension array which accumulates for each ATM model and 30 Day Block of time the loss quantities. These values are simply assigned the individual model loss quantities for the current 30 Day Block. The unit of measure is ATM.
TKAVG APAVG ATAVR HEAVG	These 4 variables are 2 dimension arrays which specify for Tanks, APC'S ATM's and Helicopters respectively, for each model type and 30 Day Block, the average daily authorization quantities. The values are derived by: dividing the cumulative totals (i.e., TKCUM, APCUM, etc.) by the number of days in the time period.
PTK PAP PHE PAT	These four variables are 2 dimensional arrays which denote for Tanks, APC's, Helicopters, and ATM's, for each model of weapon in a 30 Day Block of time the daily fractional loss rate. These values are derived by dividing summation of the current 30 Day Block and all preceding blocks' cumulative losses (TKLCUM, - APLCUM, etc.) by the average daily authorization quantities multiplied by the number of the 30 Day Block of time being analyzed.
AVGTK AVGAP AVGHE AVGAT	These four variables are arrays with 2 dimensions. Entries in these arrays specify for each model of each type of equipment for each 30 Day Block of the average daily authorization quantities. These values are derived by dividing the cumulative authorization levels (e.g., TKCUM, APCUM, etc.) by 30.
TKPCT APPCT HEPCT ATPCT	These four variables are arrays with 2 dimensions. Entries in these arrays denote for each model of each of the 4 major types of equipment for each 30 Day Block of time the average daily loss rates. These values are derived by dividing the cumulative losses (e.g., TKLCUM, APLCUM, etc) by average daily authorization levels (e.g., AVGTK, AVGAP, etc.)

TK15,  
AP15,  
AT15,  
HE15,

These four variables are single dimension arrays. Each array has sufficient occurrences to accommodate the various models of each weapon type; i.e., tank, APC, Helicopters, or ATM. Each entry specifies the average daily authorization quantities for the first 30 days of the study. This value is derived by accumulating the authorization quantities for the first 30 days as read from the CEM LOG REPORT and dividing it by 30.

TKL15,  
APL15,  
ATL15,  
HEL15,

These four variables are single dimension arrays. Each has sufficient occurrences to accommodate the various models of each weapon type. Entries in these arrays denote for the first 30 days of the study the average daily loss rates. These values are derived by accumulating losses for each weapon model for the first 30 days and dividing the result by the average daily authorization quantity.

TK30  
AP30  
AT30  
HE30  
TKL30  
APL30  
ATL30  
HEL30

These 4 variables are similar in structure and deviation to those discussed above except they denote daily authorization quantities and loss rates for the second 30 day period.

ITC

An integer variable which denotes the number of theater cycles being played. A theater cycle is equal to 4 days. This value is provided manually by the user and is read by the utility using logical unit 5.

KDC

An integer variable which denotes the number of Division Cycles. This value is equal to 8 times the number of theater cycles (ITC).

KCC

An integer variable which denotes the number of Corps Cycles. This value is equal 4 times the number of theater cycles (ITC).

K30

An integer variable which denotes the number of 30 day periods of increments in the study. This value is equal to KCC/30.

KO

An integer variable used as a counter to identify the particular 30 day Block time period being analyzed.

J, JJ

Integer variables used as counters and as subscripts in various arrays to identify specific weapon models in the 4 major weapon types.

I, IJ, IKO, II

Various integer variables used as subscripts in arrays in the program.

**15.2 OPERATING ENVIRONMENT:** This program is implemented on the EXECUTIVE-8 operating system.

**15.2.1 SUPPORT SOFTWARE:** This utility is written in and requires the FORTRAN IV compiler. In addition it also requires the UNIVAC EXEC 8 system facilities.

**15.2.2 I/O DEVICES:** The utility uses as input a file which resides on disk and produces as output a file which will reside on disk.

**15.3 MAINTENANCE PROCEDURES:** This program is normally maintained on the system by the MPP analyst.

**15.3.1 PROGRAMMING CONVENTIONS:** One unique facet of this utility is that it and the following routine, CEM/LOSSES are executed from the same runstream. Secondly, approximately 20% of the code has been added since the original routine was put into service. These additions are clearly preceded by the comment "HANK CHANGES" and terminated by "END HANK CHANGES". The variable names used in the utility are rather cryptic and heavy reliance on the variable dictionary is anticipated. The utility is structured into several general sections. There is an initialization section where various arrays are initialized to zero, the input file is read and selected arrays are filled. The next several sections calculate data for time periods INTENSE and SUSTAINING periods. Finally the results are written to the output file.

**15.2.2 INTERNAL ERROR ROUTINES:** There are no explicit error handling routines written into this utility. As a result detection of errors will depend upon the system and the user will be provided with system error messages in order to identify and solve problems.

# CEMIDATA STRUCTURE

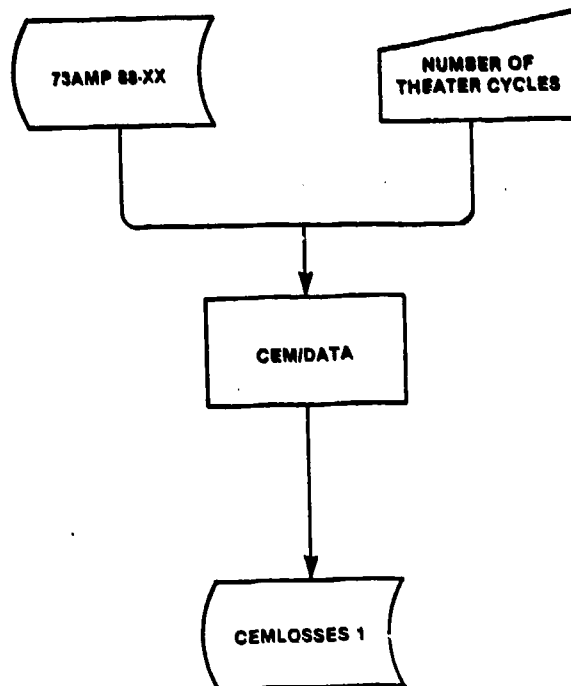


Figure III.15.1

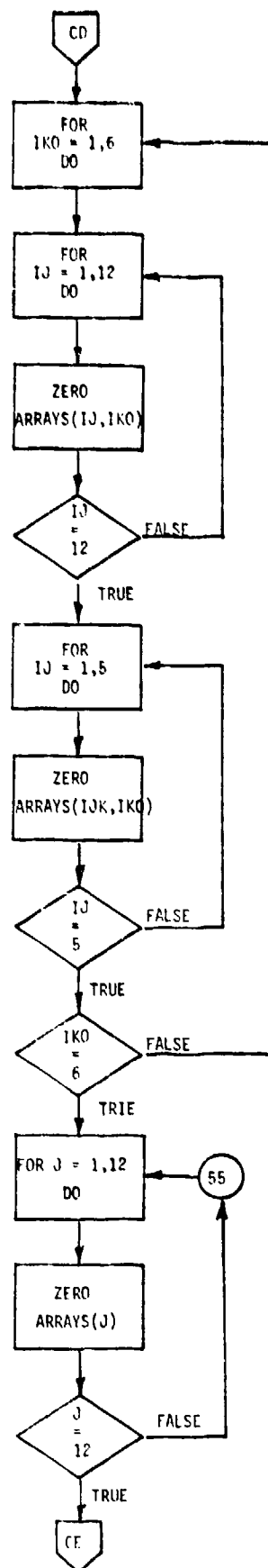
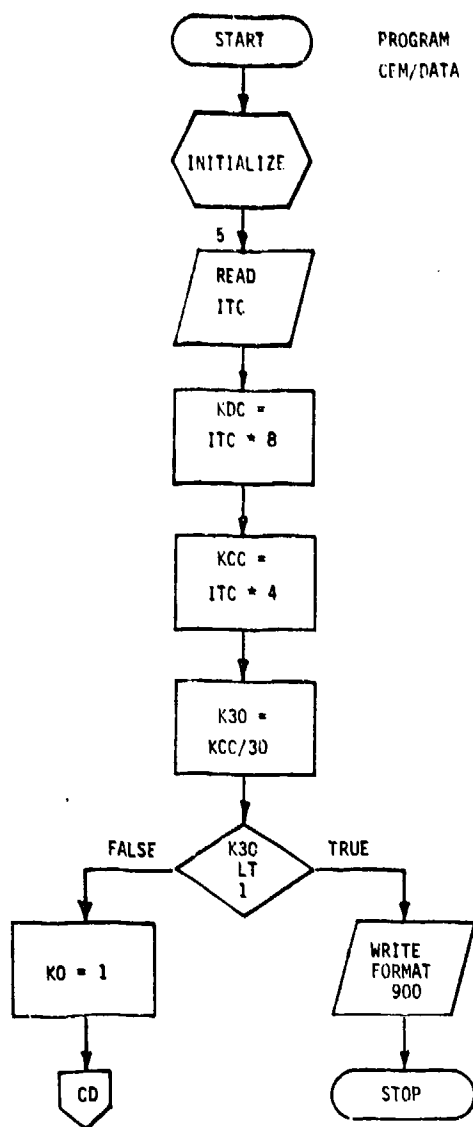
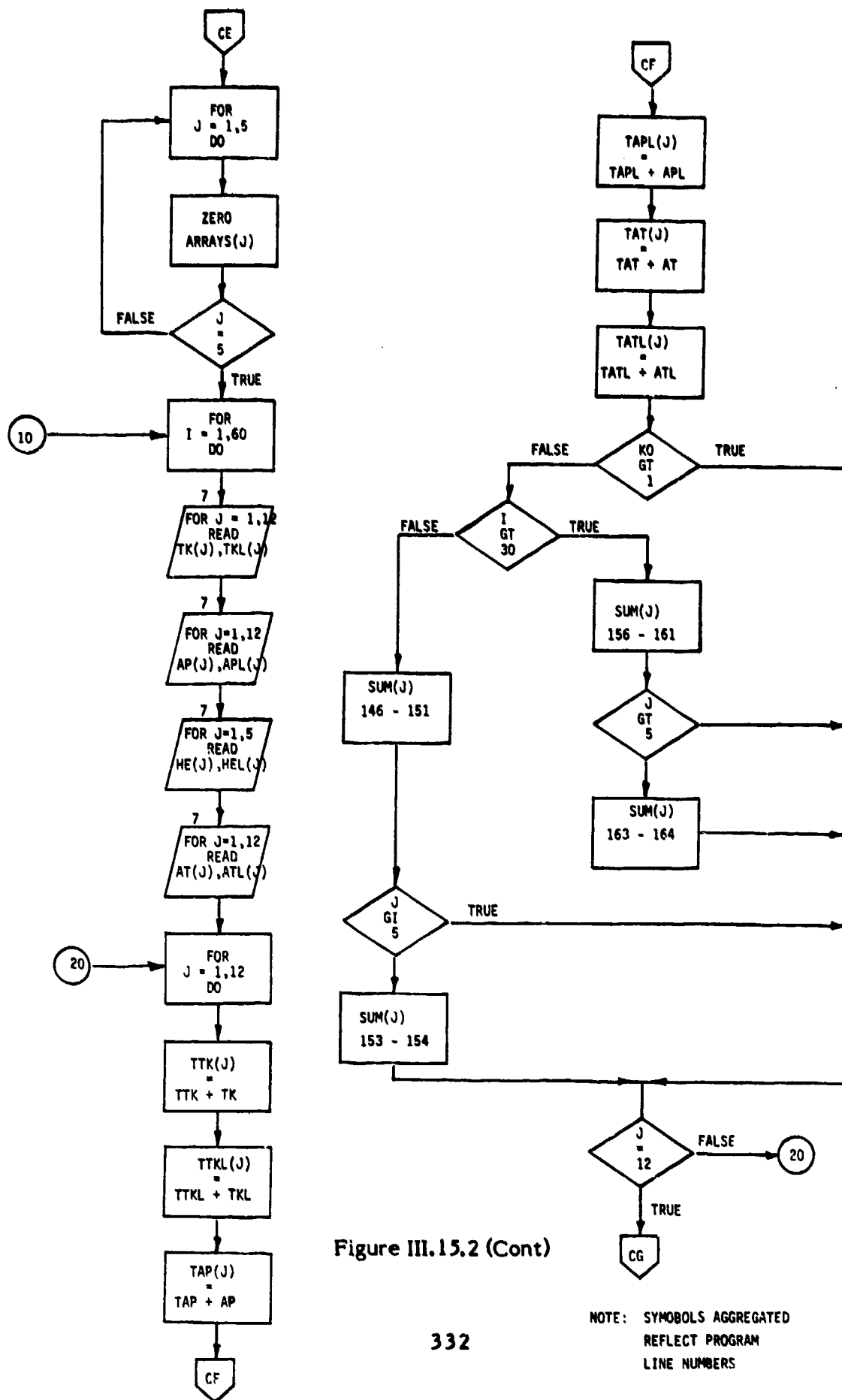
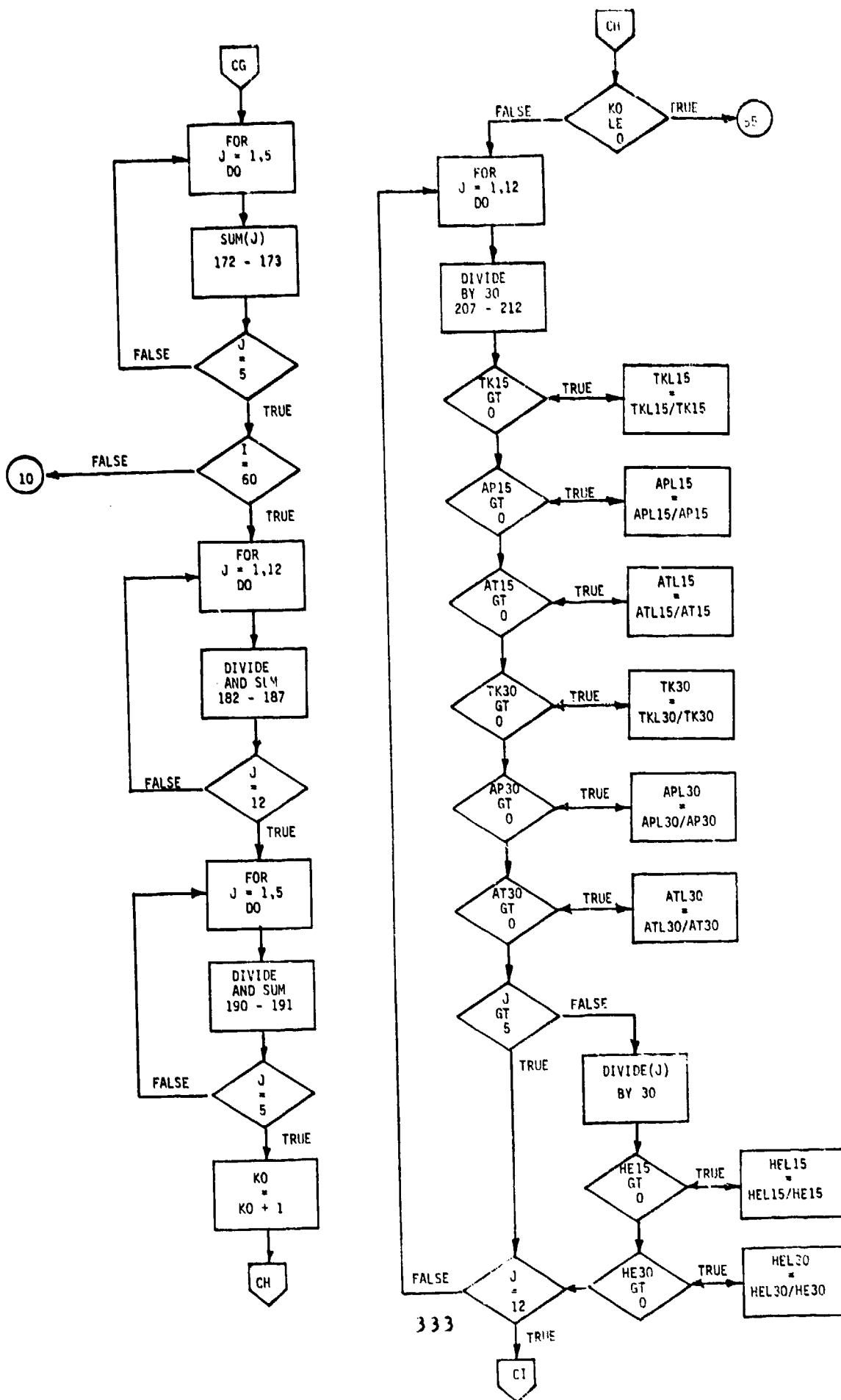
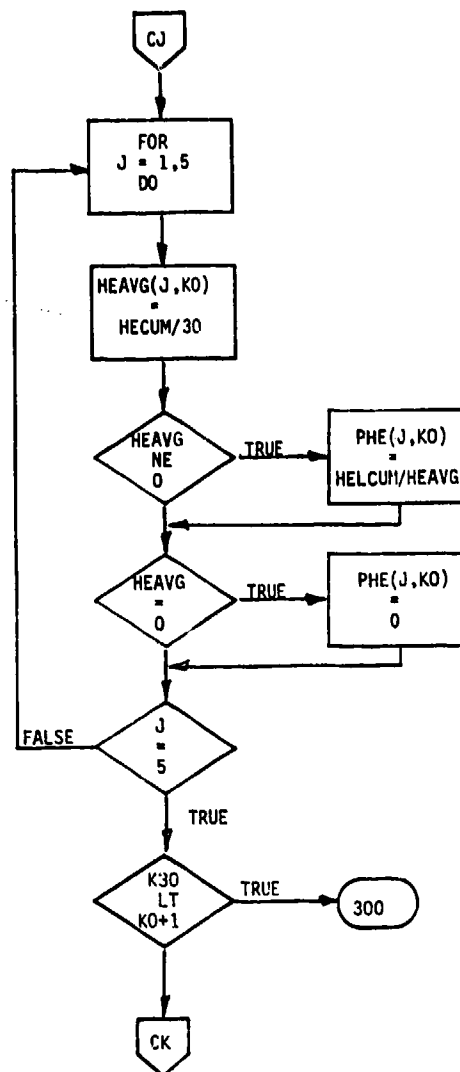
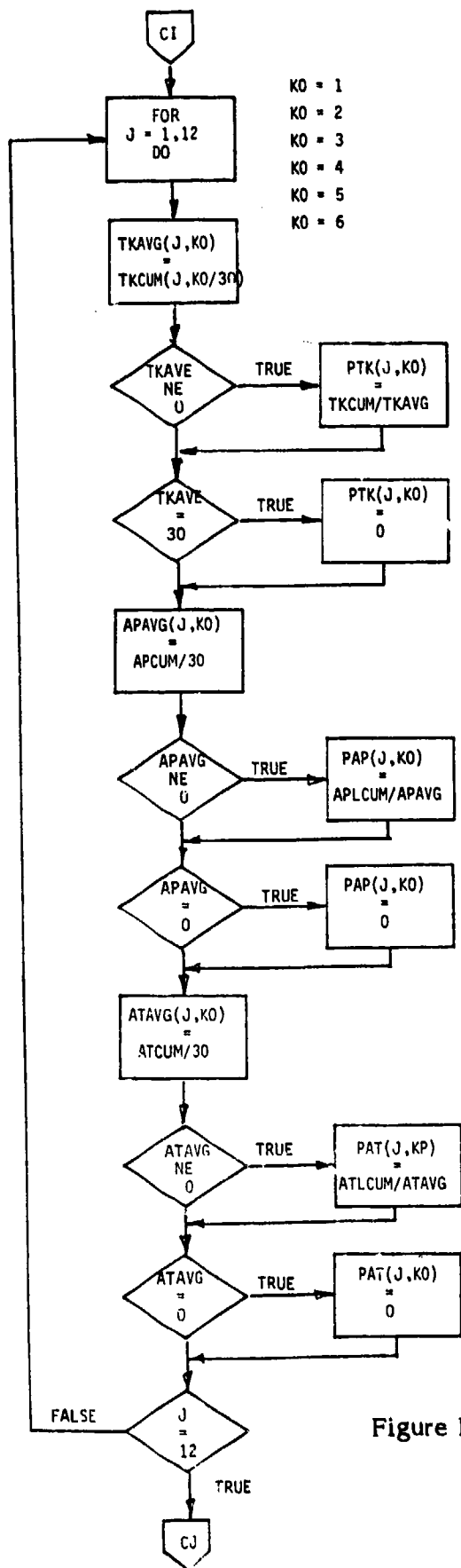


Figure III.15.2

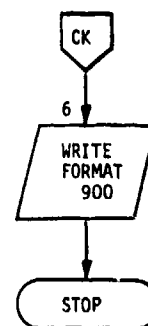






The maintenance programmer should note that this Flow-Charted pattern is repeated in the source code for each 30-day period; detail checks of the computations performed are necessary.

Figure III.15.2 (Cont)



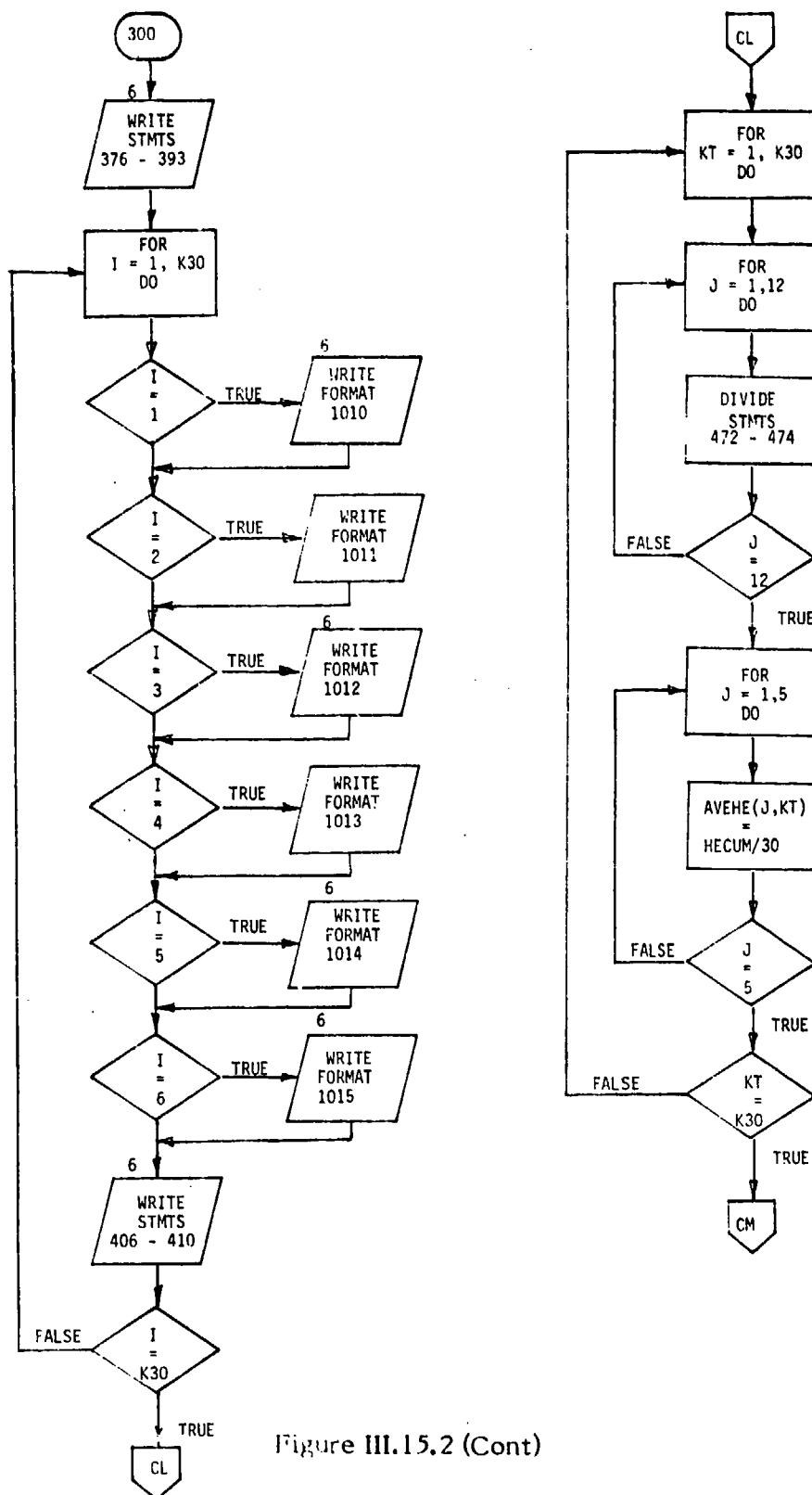
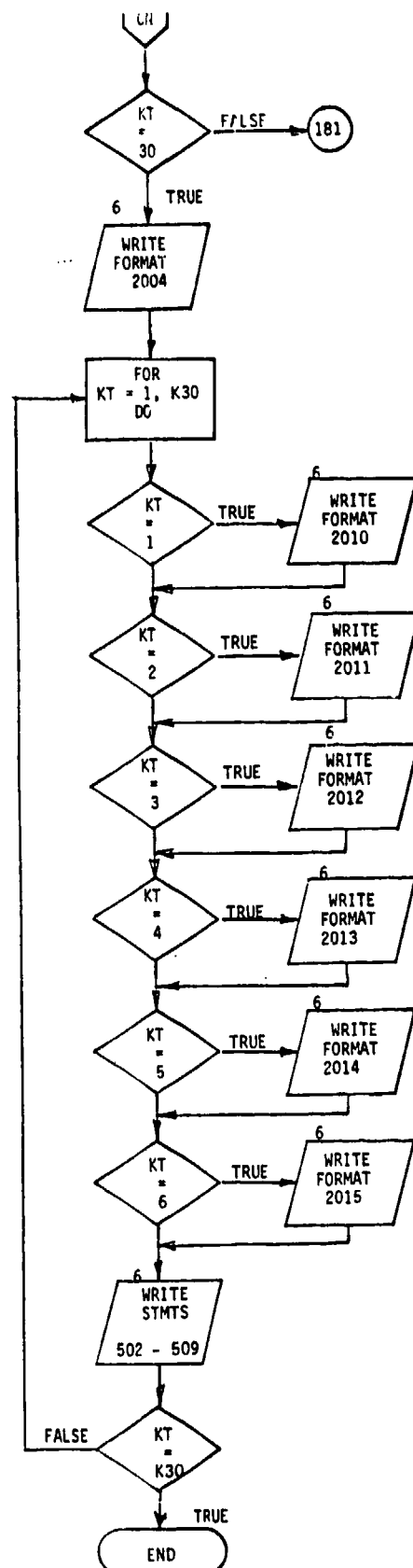
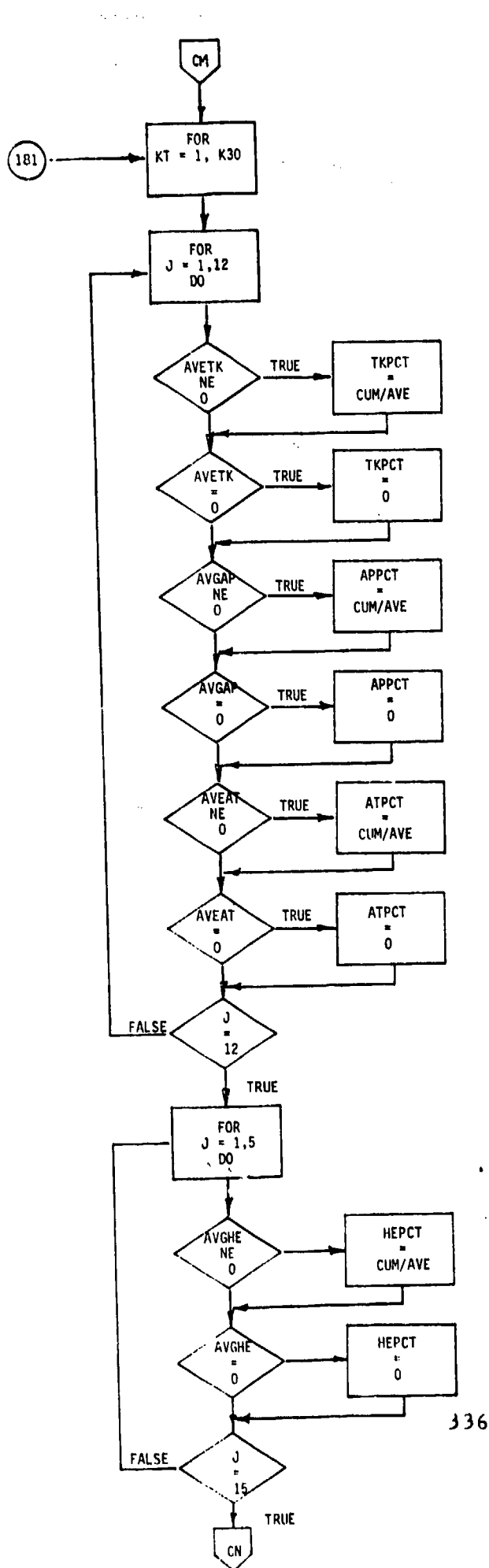


Figure III.15.2 (Cont)



UNCLASSIFIED\*\*FILE NAME:82XQT ELEMENT NAME:CEM/DATA\*\*UNCLASSIFIED

```

1:C*****
2:C*****
3:C*****THIS UTILITY IS DESIGNED TO READ THE CEM DATA REPORT
4:C*****PREPARED EXCLUSIVELY FOR WARP FROM THE CEM LOG REPORT
5:C***** (LOGREP) AND PROVIDES THE INPUT NEEDED FOR THE UTILITY
6:C*****"82XQT.CEM/LOSSES".
7:C*****THIS UTILITY AND "82XQT.CEM/LOSSES ARE EXECUTED WITHIN
8:C*****THE SAME RUN STRFAM. ONLY THE OUTPUT FROM THE LATTER
9:C*****UTILITY IS RETAINED.
10:C*****
11:C*****
12:C*****DIMENSION THE VARIABLES
13:C*****
14:C*****    DIMENSION TK(12),TKL(12),AP(12),APL(12),HE(5),HEL(5),
15:C*****        AT(12),ATL(12),
16:C*****        TTK(12),TTKL(12),TAP(12),TAPL(12),THE(5),THEL(5),
17:C*****        TAT(12),TATL(12),
18:C*****        TKCUM(12,6),TKLCUM(12,6),APCUM(12,6),APLCUM(12,6),
19:C*****        ATCUM(12,6),ATLCUM(12,6),HECUM(5,6),HELCUM(5,6),
20:C*****        TKAVG(12,6),PTK(12,6),APAVS(12,6),PAP(12,6),
21:C*****        HEAVG(5,6),PHE(5,6),ATAVG(12,6),PAT(12,6),
22:C*****        TKPCT(12,6),APPCT(12,6),HEPCT(5,6),ATPCT(12,6),
23:C*****        AVGTK(12,6),AVGAP(12,6),AVGHE(5,6),AVGAT(12,6)
24:C*****
25:C*****HANK CHANGES
26:C*****
27:C*****    DIMENSION TK15(12),TK30(12),AP15(12),AP30(12),AT15(12),AT30(12),
28:C*****        TKL15(12),TKL30(12),APL15(12),APL30(12),ATL15(12),ATL30(12),
29:C*****        HE15(5),HE30(5),HEL15(5),HEL30(5)
30:C*****
31:C*****END CHANGES
32:C*****
33:C*****    READ (5,1000) ITC
34:C*****    KDC = ITC*8
35:C*****    KCC = ITC*4
36:C*****    K30 = KCC/30
37:C*****    IF(K30.LT.1) GO TO 200
38:C*****
39:C*****KDC = NUMBER OF DIVISION CYCLES KCC = NUMBER OF CORPS CYCLES
40:C*****K30 = NUMBER OF 30-DAY INCREMENTS .
41:C*****TK = TANK, AP = APC, HE = HELO, AT = ANTI-TANK & MORTARS.
42:C*****TKL, APL, HEL, ATL = 12-HOUR LOSSES OF ITEMS OF EQUIPMENT.
43:C*****TTK, TAP, THE, TAT = CUMULATIVE NUMBER OF ITEMS OVER 30 DAYS.
44:C*****TTKL, TTAPL, TTHEL, TTATL = CUMULATIVE NUMBER OF ITEMS LOST OVER
45:C*****A 30 DAY PERIOD.
46:C*****
47:C*****
48:C*****TKCUM, APCUM, HECUM, ATCUM = CUMULATIVE NUMBER OF ITEMS BY 30
49:C*****DAY BLOCK.
50:C*****
51:C*****TKLCUM, APLCUM, HELCUM, ATLCUM = CUMULATIVE LOSS BY 30 DAY
52:C*****BLOCK.
53:C*****
54:C*****
55:C*****J OR JJ IS ALWAYS USED AS THE COUNTER FOR TYPE OF WEAPON
56:C*****IN A MAJOR CATEGORY.
57:C*****

```

Figure III.15.3

UNCLASSIFIED\*\*\*FILE NAME:82XQT ELEMENT NAME:CEM/DATA\*\*\*UNCLASSIFIED

```

58:C*****
59:C*****KD IS THE COUNTER FOR EACH 30 DAY BLOCK OF DATA.
60:C*****
61:C*****
62:C*****
63:      KD = 1
64:C*****
65:C*****
66:C*****ZERO THE ARRAYS
67:C*****
68:C*****
69:      1 DO 2 IKD= 1,6
70:      DO 3 IJ = 1,12
71:      TKCUM(IJ,IKD) = 0.
72:      APCUM(IJ,IKD) = 0.
73:      ATCUM(IJ,IKD) = 0.
74:      TKLCUM(IJ,IKD) = 0.
75:      APLCUM(IJ,IKD) = 0.
76:      ATLCUM(IJ,IKD) = 0.
77:      TKAVG(IJ,IKD) = 0.
78:      APAVG(IJ,IKD) = 0.
79:      ATAVG(IJ,IKD) = 0.
80:      PTK(IJ,IKD) = 0.
81:      PAP(IJ,IKD) = 0.
82:      PAT(IJ,IKD) = 0.
83:      3 CONTINUE
84:      DO 4 IJ = 1,5
85:      HECUM(IJ,IKD) = 0.
86:      HELCUM(IJ,IKD) = 0.
87:      HEAVG(IJ,IKD) = 0.
88:      PHE(IJ,IKD) = 0.
89:      4 CONTINUE
90:      2 CONTINUE
91:C*****
92:C*****
93:C*****ZERO THE AUTHORIZED AND LOSS DATA EVERY 30-DAY PERIOD.
94:C*****EACH 30-DAY PERIOD IS EQUIVALENT TO 60 DIVISION CYCLES.
95:C*****
96:C*****
97:      55 DO 60 J = 1,12
98:      TK(J) = 0.
99:      TKL(J) = 0.
100:      AP(J) = 0.
101:      APL(J) = 0.
102:      AT(J) = 0.
103:      ATL(J) = 0.
104:      TTK(J) = 0.
105:      TTKL(J) = 0.
106:      TAP(J) = 0.
107:      TAPL(J) = 0.
108:      TAT(J) = 0.
109:      TATL(J) = 0.
110:      60 CONTINUE
111:      DO 70 J = 1,5
112:      HE(J) = 0.
113:      HEL(J) = 0.
114:      THE(J) = 0.

```

Figure III.15.3 (Cont)

UNCLASSIFIED\*\*\*FILE NAME:8ZXQT ELEMENT NAME:CEM/DATA\*\*\*UNCLASSIFIED

```

115:      THEL(J) = 0.
116:      70 CONTINUE
117:      DO 10 I = 1,60
118:C*****
119:C*****
120:C*****READ THE AUTHORIZED AND LOSS DATA FOR EACH DIVISION CYCLE.
121:C*****
122:C*****
123:      READ(7,1001) (TK(J), TKL(J), J=1,12)
124:      READ(7,1001) (AP(J), APL(J), J=1,12)
125:      READ(7,1002) (HE(J), HEL(J), J=1,5)
126:      READ(7,1001) (AT(J), ATL(J), J=1,12)
127:C*****
128:C*****
129:C*****ACCUMULATE THE AUTHORIZED AND LOSS DATA FOR THE CURRENT 30 DAYS.
130:C*****
131:C*****
132:      DO 20 J = 1,12
133:      TTK(J) = TTK(J) + TK(J)
134:      TTKL(J) = TTKL(J) + TKL(J)
135:      TAP(J) = TAP(J) + AP(J)
136:      TAPL(J) = TAPL(J) + APL(J)
137:      TAT(J) = TAT(J) + AT(J)
138:      TATL(J) = TATL(J) + ATL(J)
139:C*****
140:C*****
141:C*****HANK CHANGES
142:C*****
143:C*****
144:      IF (KD.CY.1) GO TO 20
145:      IF (I.GT.30) GO TO 18
146:      TK15 (J) = TK15 (J) + TK (J)
147:      AP15 (J) = AP15 (J) + AP (J)
148:      AT15 (J) = AT15 (J) + AT (J)
149:      TKL15 (J) = TKL15 (J) + TKL (J)
150:      APL15 (J) = APL15 (J) + APL (J)
151:      ATL15 (J) = ATL15 (J) + ATL (J)
152:      IF (J.GT.5) GO TO 20
153:      HE15 (J) = HE15 (J) + HE (J)
154:      HEL15 (J) = HEL15 (J) + HEL (J)
155:      GO TO 20
156: 18      TK30 (J) = TK30 (J) + TK (J)
157:      AP30 (J) = AP30 (J) + AP (J)
158:      AT30 (J) = AT30 (J) + AT (J)
159:      TKL30 (J) = TKL30 (J) + TKL (J)
160:      APL30 (J) = APL30 (J) + APL (J)
161:      ATL30 (J) = ATL30 (J) + ATL (J)
162:      IF (J.GT.5) GO TO 20
163:      HE30 (J) = HE30 (J) + HE (J)
164:      HEL30 (J) = HEL30 (J) + HEL (J)
165:C*****
166:C*****
167:C*****END HANK CHANGES
168:C*****
169:C*****
170:      20 CONTINUE
171:      DO 30 J = 1,5

```

Figure III.15.3 (Cont)

UNCLASSIFIED\*\*\*FILE NAME:82XOT ELEMENT NAME:CEM/DATA\*\*\*UNCLASSIFIED

```

172:     THE(J) = THE(J) + HE(J)
173:     THEL(J) = THEL(J) + HEL(J)
174:     30 CONTINUE
175:     10 CONTINUE
176:C*****
177:C*****
178:C*****ACCUMULATE AUTHORIZED AND LOSS DATA BY 30 DAY BLOCK (KO).
179:C*****
180:C*****
181:     DO 40 J = 1,12
182:         TKCUM(J,KO) = TTK(J)/2.          D(60 DIV CYC)/2=CUM FOR 30 DAYS.
183:         TKLCUM(J,KO) = TTKL(J)
184:         APCUM(J,KO) = TAP(J)/2.
185:         APLCUM(J,KO) = TAPL(J)
186:         ATCUM(J,KO) = TAT(J)/2.
187:         ATLCUM(J,KO) = TATL(J)
188:     40 CONTINUE
189:     DO 50 J = 1,5
190:         HECUM(J,KO) = THE(J)/2.
191:         HELCUM(J,KO) = THEL(J)
192:     50 CONTINUE
193:C*****
194:C*****
195:C*****INCREMENT THE 30-DAY PERIOD COUNTER (KO).
196:C*****
197:C*****
198:     KO = KO + 1
199:     IF(KO.LE.K30) GO TO 55
200:C*****
201:C*****
202:C*****HANK CHANGE
203:C*****
204:C*****
205:     DO 57 J=1,12
206:         THIRTY=30.
207:         TK15 (J) = TK15 (J)/THIRTY
208:         AP15 (J) = AP15 (J)/THIRTY
209:         AT15 (J) = AT15 (J)/THIRTY
210:         TK30 (J) = TK30 (J)/THIRTY
211:         AP30 (J) = AP30 (J)/THIRTY
212:         AT30 (J) = AT30 (J)/THIRTY
213:         IF(TK15(J).GT.0.) TKL15 (J) = TKL15 (J)/ TK15 (J)
214:         IF(AP15(J).GT.0.) APL15 (J) = APL15 (J)/ AP15 (J)
215:         IF(AT15(J).GT.0.) ATL15 (J) = ATL15 (J)/ AT15 (J)
216:         IF(TK30(J).GT.0.) TKL30 (J) = TKL30 (J)/ TK30 (J)
217:         IF(AP30(J).GT.0.) APL30 (J) = APL30 (J)/ AP30 (J)
218:         IF(AT30(J).GT.0.) ATL30 (J) = ATL30 (J)/ AT30 (J)
219:         IF(J.GT.5) GO TO 57
220:         HE15 (J) = HE15 (J)/THIRTY
221:         HE30 (J) = HE30 (J)/THIRTY
222:         IF(HE15(J).GT.0.) HEL15 (J) = HEL15 (J)/ HE15 (J)
223:         IF(HE30(J).GT.0.) HEL30 (J) = HEL30 (J)/ HE30 (J)
224:     57 CONTINUE
225:C*****
226:C*****
227:C*****END HANK CHANGES
228:C*****CALCULATE DATA FOR 1ST 30-DAY PERIOD (KO=1) *INTENSE PD*

```

Figure III.15.3 (Cont)

UNCLASSIFIED\*\*\*FILE NAME:82XQT ELEMENT NAME:CEM/DATA\*\*\*UNCLASSIFIED

```

229:C*****
230:C*****
231: 110 DO 111 J = 1,12
232:     TKAVG(J,1) = TKCUM(J,1)/30.
233:     IF(TKAVG(J,1).NE.0)PTK(J,1) = TKLCUM(J,1)/TKAVG(J,1)
234:     IF(TKAVG(J,1).EQ.0)PTK(J,1) = 0
235:     APAVG(J,1) = APCUM(J,1)/30.
236:     IF(APAVG(J,1).NE.0)PAP(J,1) = APLCUM(J,1)/APAVG(J,1)
237:     IF(APAVG(J,1).EQ.0)PAP(J,1) = 0
238:     ATAVG(J,1) = ATCUM(J,1)/30.
239:     IF(ATAVG(J,1).NE.0)PAT(J,1) = ATLCUM(J,1)/ATAVG(J,1)
240:     IF(ATAVG(J,1).EQ.0)PAT(J,1) = 0
241: 111 CONTINUE
242:     DO 112 J = 1,5
243:     HEAVG(J,1) = HECUM(J,1)/30.
244:     IF(HEAVG(J,1).NE.0)PHE(J,1) = HELCUM(J,1)/HEAVG(J,1)
245:     IF(HEAVG(J,1).EQ.0)PHE(J,1) = 0
246: 112 CONTINUE
247:     IF (K30.LT.2) GO TO 300
248:C*****
249:C*****
250:C****CALCULATE DATA FOR 2ND 30-DAY PERIOD (KQ=2) *INTENSE PD*
251:C*****
252:C*****
253: 120 DO 121 J = 1,12
254:     TKAVG(J,2) = (TKCUM(J,1) + TKCUM(J,2))/60.
255:     IF(TKAVG(J,2).NE.0)PTK(J,2) = (TKLCUM(J,1) + TKLCUM(J,2))/
256:     * (TKAVG(J,2)*2.)
257:     IF(TKAVG(J,2).EQ.0)PTK(J,2) = 0
258:     APAVG(J,2) = (APCUM(J,1) + APCUM(J,2))/60.
259:     IF(APAVG(J,2).NE.0)PAP(J,2) = (APLCUM(J,1) + APLCUM(J,2))/
260:     * (APAVG(J,2)*2.)
261:     IF(APAVG(J,2).EQ.0)PAP(J,2) = 0
262:     ATAVG(J,2) = (ATCUM(J,1) + ATCUM(J,2))/60.
263:     IF(ATAVG(J,2).NE.0)PAT(J,2) = (ATLCUM(J,1) + ATLCUM(J,2))/
264:     * (ATAVG(J,2)*2.)
265:     IF(ATAVG(J,2).EQ.0)PAT(J,2) = 0
266: 121 CONTINUE
267:     DO 122 J = 1,5
268:     HEAVG(J,2) = (HECUM(J,1) + HECUM(J,2))/60.
269:     IF(HEAVG(J,2).NE.0)PHE(J,2) = (HELCUM(J,1) + HELCUM(J,2))/
270:     * (HEAVG(J,2)*2.)
271:     IF(HEAVG(J,2).EQ.0)PHE(J,2) = 0
272: 122 CONTINUE
273:     IF (K30.LT.3) GO TO 300
274:C*****
275:C*****
276:C****CALCULATE DATA FOR 3RD 30-DAY PERIOD (KQ=3) *INTENSE PD*
277:C*****
278:C*****
279: 130 DO 131 J = 1,12
280:     TKAVG(J,3) = (TKCUM(J,1) + TKCUM(J,2) + TKCUM(J,3))/90.
281:     IF(TKAVG(J,3).NE.0)PTK(J,3) = (TKLCUM(J,1) + TKLCUM(J,2) +
282:     * TKLCUM(J,3)) / (TKAVG(J,3)*3.)
283:     IF(TKAVG(J,3).EQ.0)PTK(J,3) = 0
284:     APAVG(J,3) = (APCUM(J,1) + APCUM(J,2) + APCUM(J,3))/90.
285:     IF(APAVG(J,3).NE.0)PAP(J,3) = (APLCUM(J,1) + APLCUM(J,2) +

```

Figure III.15.3 (Cont)

UNCLASSIFIED\*\*\*FILE NAME:82XQT ELEMENT NAME:CEM/DATA\*\*\*UNCLASSIFIED

```

286:      *APLCUM(J,3))/(APAVG(J,3)*3.)
287:      IF(APAVG(J,3).EQ.0)PAP(J,3) = 0
288:      ATAVG(J,3) = (ATCUM(J,1) + ATCUM(J,2) + ATCUM(J,3))/90.
289:      IF(ATAVG(J,3).NE.0)PAT(J,3) = (ATLCUM(J,1)+ ATLCUM(J,2)+
290:      *ATLCUM(J,3))/(ATAVG(J,3)*3.)
291:      IF(ATAVG(J,3).EQ.0)PAT(J,3) = 0
292: 131 CONTINUE
293:      DO 132 J = 1,5
294:      HEAVG(J,3) = (HECUM(J,1) + HECUM(J,2) + HECUM(J,3))/90.
295:      IF(HEAVG(J,3).NE.0)PHE(J,3) = (HELUM(J,1)+ HELUM(J,2)+
296:      *HELUM(J,3))/(HEAVG(J,3)*3.)
297:      IF(HEAVG(J,3).EQ.0)PHE(J,3) = 0
298: 132 CONTINUE
299:      IF (K30.LT.4) GO TO 300
300:C*****
301:C*****
302:C*****CALCULATE DATA FOR 4TH 30-DAY PERIOD (KQ=4) *SUSTAINING PD*
303:C*****
304:C*****
305: 140 DO 141 J = 1,12
306:      TKAVG(J,4) = TKCUM(J,4)/30.
307:      IF(TKAVG(J,4).NE.0)PTK(J,4) = TKLCUM(J,4)/TKAVG(J,4)
308:      IF(TKAVG(J,4).EQ.0)PTK(J,4) = 0
309:      APAVG(J,4) = APCUM(J,4)/30.
310:      IF(APAVG(J,4).NE.0)PAP(J,4) = APLCUM(J,4)/APAVG(J,4)
311:      IF(APAVG(J,4).EQ.0)PAP(J,4) = 0
312:      ATAVG(J,4) = ATCUM(J,4)/30.
313:      IF(ATAVG(J,4).NE.0)PAT(J,4) = ATLCUM(J,4)/ATAVG(J,4)
314:      IF(ATAVG(J,4).EQ.0)PAT(J,4) = 0
315: 141 CONTINUE
316:      DO 142 J = 1,5
317:      HEAVG(J,4) = HECUM(J,4)/30.
318:      IF(HEAVG(J,4).NE.0)PHE(J,4) = HELCUM(J,4)/HEAVG(J,4)
319:      IF(HEAVG(J,4).EQ.0)PHE(J,4) = 0
320: 142 CONTINUE
321:      IF (K30.LT.5) GO TO 300
322:C*****
323:C*****
324:C*****CALCULATE DATA FOR 5TH 30-DAY PERIOD (KQ=5) *SUSTAINING PD*
325:C*****
326:C*****
327: 150 DO 151 J = 1,12
328:      TKAVG(J,5) = (TKCUM(J,4) + TKCUM(J,5))/60.
329:      IF(TKAVG(J,5).NE.0)PTK(J,5) = (TKLCUM(J,4) + TKLCUM(J,5))/
330:      *(TKAVG(J,5)*2.)
331:      IF(TKAVG(J,5).EQ.0)PTK(J,5) = 0
332:      APAVG(J,5) = (APCUM(J,4) + APCUM(J,5))/60.
333:      IF(APAVG(J,5).NE.0)PAP(J,5) = (APLCUM(J,4) + APLCUM(J,5))/
334:      *(APAVG(J,5)*2.)
335:      IF(APAVG(J,5).EQ.0)PAP(J,5) = 0
336:      ATAVG(J,5) = (ATCUM(J,4) + ATCUM(J,5))/60.
337:      IF(ATAVG(J,5).NE.0)PAT(J,5) = (ATLCUM(J,4) + ATLCUM(J,5))/
338:      *(ATAVG(J,5)*2.)
339:      IF(ATAVG(J,5).EQ.0)PAT(J,5) = 0
340: 151 CONTINUE
341:      DO 152 J = 1,5
342:      HEAVG(J,5) = (HECUM(J,4) + HECUM(J,5))/60.

```

Figure III.15.3 (Cont)

UNCLASSIFIED\*\*\*FILE NAME:82XGT ELEMENT NAME:CEM/DATA\*\*\*UNCLASSIFIED

```

343:      IF(HEAVG(J,5).NE.0)PHE(J,5) = (HELCUM(J,4) + HELCUM(J,5))/
344:      * (HEAVG(J,5)*2.)
345:      IF(HEAVG(J,5).EQ.0)PHE(J,5) = 0
346: 152 CONTINUE
347:      IF (K30.LT.6) GO TO 300
348:C*****
349:C*****
350:C*****CALCULATE DATA FOR 6TH 30-DAY PERIOD (K0=6) *SUSTAINING PD*
351:C*****
352:C*****
353: 160 DO 161 J = 1,12
354:      TKAVG(J,6) = (TKCUM(J,4) + TKCUM(J,5) + TKCUM(J,6))/90.
355:      IF(TKAVG(J,6).NE.0)PTK(J,6) = (TKLCUM(J,4)+ TKLCUM(J,5)+
356:      *TKLCUM(J,6))/(TKAVG(J,6)*3.)
357:      IF(TKAVG(J,6).EQ.0)PTK(J,6) = 0
358:      APAVG(J,6) = (APCUM(J,4) + APCUM(J,5) + APCUM(J,6))/90.
359:      IF(APAVG(J,6).NE.0)PAP(J,6) = (APLCUM(J,4)+ APLCUM(J,5)+
360:      *APLCUM(J,6))/(APAVG(J,6)*3.)
361:      IF(APAVG(J,6).EQ.0)PAP(J,6)=0
362:      ATAVG(J,6) = (ATCUM(J,4) + ATCUM(J,5) + ATCUM(J,6))/90.
363:      IF(ATAVG(J,6).NE.0)PAT(J,6) = (ATLCUM(J,4)+ ATLCUM(J,5)+
364:      *ATLCUM(J,6))/(ATAVG(J,6)*3.)
365:      IF(ATAVG(J,6).EQ.0)PAT(J,6) = 0
366: 161 CONTINUE
367:      DO 162 J = 1,5
368:      HEAVG(J,6) = (HECUM(J,4) + HECUM(J,5) + HECUM(J,6))/90.
369:      IF(HEAVG(J,6).NE.0)PHE(J,6) = (HELCUM(J,4)+ HELCUM(J,5)+
370:      *HELCUM(J,6))/(HEAVG(J,6)*3.)
371:      IF(HEAVG(J,6).EQ.0)PHE(J,6) = 0
372: 162 CONTINUE
373:      IF (K30.GE.1) GO TO 300
374: 200 WRITE(6,900)
375:      STOP
376: 300 WRITE(6,1004) K30
377:C*****
378:C*****
379:C*****HANK CHANGES
380:C*****
381:C*****
382:      WRITE(6,3010)
383:      WRITE(6,1003) (II,I=1,12)
384:      WRITE(6,1005) TK15,TKL15
385:      WRITE(6,1006) AP15,APL15
386:      WRITE(6,1007) HE15,HEL15
387:      WRITE(6,1008) AT15,ATL15
388:      WRITE(6,3011)
389:      WRITE(6,1003) (II,I=1,12)
390:      WRITE(6,1005) TK30,TKL30
391:      WRITE(6,1006) AP30,APL30
392:      WRITE(6,1007) HE30,HEL30
393:      WRITE(6,1008) AT30,ATL30
394:C*****
395:C*****
396:C*****END HANK CHANGES
397:C*****
398:C*****
399: 400 DO 401 I=1,K30

```

Figure III.15.3 (Cont)

UNCLASSIFIED\*\*\*FILE NAME:02XGT ELEMENT NAME:CEM/DATA\*\*\*UNCLASSIFIED

```

400:      IF(I.EQ.1) WRITE(6,1010)
401:      IF(I.EQ.2) WRITE(6,1011)
402:      IF(I.EQ.3) WRITE(6,1012)
403:      IF(I.EQ.4) WRITE(6,1013)
404:      IF(I.EQ.5) WRITE(6,1014)
405:      IF(I.EQ.6) WRITE(6,1015)
406:      WRITE(6,1003) (II,II=1,12)
407:      WRITE(6,1005) ((TKAVG(J,I),J=1,12),(PTK(J,I),JJ=1,12))
408:      WRITE(6,1006) ((APAVG(J,I),J=1,12),(PAF(J,I),JJ=1,12))
409:      WRITE(6,1007) ((HEAVG(J,I),J=1,5),(PHE(J,I),JJ=1,5))
410:      WRITE(6,1008) ((ATAVG(J,I),J=1,12),(PAT(J,I),JJ=1,12))
411:  401 CONTINUE
412:  900 FORMAT(1H 'THTR CYCLES LESS THAN 1 30-DAY PERIOD')
413:  1000 FORMAT(36X,14)
414:  1001 FORMAT(3F10.4,/,8F10.4,/,9F10.4)
415:  1002 FORMAT(9F10.4,/,2F10.4)
416:  1003 FORMAT(1H 'EQUIP TYPE ',12(3X,I2,3X))
417:  1004 FORMAT(1H 'NUMBER OF 30-DAY BLOCKS OF DATA = ',I2,///)
418:  1005 FORMAT(1H 'AVG TK AUTH',12F8.2,/,1H 'PCT TK LOSS',12F8.2,///)
419:  1006 FORMAT(1H 'AVG APC AUTH',12F8.2,/,1H 'PCT APC LOSS',12F8.2,///)
420:  1007 FORMAT(1H 'AVG HEL AUTH',5F8.2,/,1H 'PCT HEL LOSS',5F8.2,///)
421:  1008 FORMAT(1H 'AVG ATM AUTH',12F8.2,/,1H 'PCT ATM LOSS',12F8.2,///)
422:  1009 FORMAT(1H '30-DAY PERIOD NUMBER ',I2,///)
423:C *****
424:C *****
425:C *****HANK CHANGES
426:C *****
427:C *****
428:  1010 FORMAT(1H 27X,'AVERAGE 30-DAY LOSS RATES **D-1 TO D-30**
429:      * INTENSE PERIOD',///)
430:  3010 FORMAT(1H 27X,'AVERAGE 15-DAY LOSS RATES **D-1 TO D-15**
431:      * INTENSE PERIOD',///)
432:  3011 FORMAT(1H 27X,'AVERAGE 15-DAY LOSS RATES **D-16 TO D-30**
433:      * INTENSE PERIOD',///)
434:C *****
435:C *****
436:C *****END HANK CHANGES
437:C *****
438:C *****
439:  1011 FORMAT(1H 27X,'AVERAGE 30-DAY LOSS RATES **D-1 TO D-60**
440:      * INTENSE PERIOD',///)
441:  1012 FORMAT(1H 27X,'AVERAGE 30-DAY LOSS RATES **D-1 TO D-90**
442:      * INTENSE PERIOD',///)
443:  1013 FORMAT(1H 27X,'AVG 30-DAY LOSS RATES **D-91 TO D-120**
444:      * SUSTAINING PERIOD',///)
445:  1014 FORMAT(1H 27X,'AVG 30-DAY LOSS RATES **D-91 TO D-150**
446:      * SUSTAINING PERIOD',///)
447:  1015 FORMAT(1H 27X,'AVG 30-DAY LOSS RATES **D-91 TO D-180**
448:      * SUSTAINING PERIOD',///)
449:  2000 FORMAT(' AVG TANK AUTH',12F8.2,/,1H 'TOT TANK LOSS',12F8.2,/,
450:      '*PCT TANK LOSS',12F8.2,///)
451:  2001 FORMAT(' AVG APC AUTH',12F8.2,/' TOT APC LOSS',12F8.2,/,
452:      '* PCT APC LOSS',12F8.2,///)
453:  2002 FORMAT(' AVG HELO AUTH',5F8.2,/' TOT HELO LOSS',5F8.2,/,
454:      '* PCT HELO LOSS',5F8.2,///)
455:  2003 FORMAT(' AVG ATM AUTH',12F8.2,/' TOT ATM LOSS',12F8.2,/,
456:      '* PCT ATM LOSS',12F8.2,///)

```

Figure III.15.3 (Cont)

UNCLASSIFIED\*\*\*FILE NAME:82XQT ELEMENT NAME:CEH/DATA\*\*\*UNCLASSIFIED

```

457:2004 FORMAT(1H 10X,'**** FOLLOWING DATA FOR EACH 30-DAY PERIOD',///)
458:2010 FORMAT(1H 27X,'30-DAY LCSS RATE ** D-1 TO D-30**',//)
459:2011 FORMAT(1H 27X,'30-DAY LCSS RATE ** D-31 TO D-60**',//)
460:2012 FORMAT(1H 27X,'30-DAY LCSS RATE ** D-61 TO D-90**',//)
461:2013 FORMAT(1H 27X,'30-DAY LCSS RATE ** D-91 TO D-120**',//)
462:2014 FORMAT(1H 27X,'30-DAY LCSS RATE ** D-121 TO D-150**',//)
463:2015 FORMAT(1H 27X,'30-DAY LCSS RATE ** D-151 TO D-180**',//)
464:C*****
465:C*****
466:C*****SECTIONS 170 AND 180 CALCULATE AND PRINT THE LOSS/PCT DATA
467:C*****FOR EACH SEPARATE 30-DAY PERIOD.
468:C*****
469:C*****
470: 170 DO 171 KT = 1,K30
471:     DO 172 J = 1,12
472:         AVGTK(J,KT) = TKCUM(J,KT)/30.
473:         AVGAP(J,KT) = APCUM(J,KT)/30.
474:         AVGAT(J,KT) = ATCUM(J,KT)/30.
475: 172 CONTINUE
476:     DO 173 J = 1,5
477:         AVGHE(J,KT) = HECUM(J,KT)/30.
478: 173 CONTINUE
479: 171 CONTINUE
480: 180 DO 181 KT = 1,K30
481:     DO 182 J = 1,12
482:         IF(AVGTK(J,KT).NE.0)TKPCT(J,KT) = TKLCUM(J,KT)/AVGTK(J,KT)
483:         IF(AVGTK(J,KT).EQ.0)TKPCT(J,KT) = 0
484:         IF(AVGAP(J,KT).NE.0)APPCT(J,KT) = APLCUM(J,KT)/AVGAP(J,KT)
485:         IF(AVGAP(J,KT).EQ.0)APPCT(J,KT) = 0
486:         IF(AVGAT(J,KT).NE.0)ATPCT(J,KT) = ATLCUM(J,KT)/AVGAT(J,KT)
487:         IF(AVGAT(J,KT).EQ.0)ATPCT(J,KT) = 0
488: 182 CONTINUE
489:     DO 183 J = 1,5
490:         IF(AVGHE(J,KT).NE.0)HEPCT(J,KT) = HELCUM(J,KT)/AVGHE(J,KT)
491:         IF(AVGHE(J,KT).EQ.0)HEPCT(J,KT) = 0
492: 183 CONTINUE
493: 181 CONTINUE
494:     WRITE(6,2004)
495:     DO 191 KT=1,K30
496:         IF(KT.EQ.1) WRITE(6,2010)
497:         IF(KT.EQ.2) WRITE(6,2011)
498:         IF(KT.EQ.3) WRITE(6,2012)
499:         IF(KT.EQ.4) WRITE(6,2013)
500:         IF(KT.EQ.5) WRITE(6,2014)
501:         IF(KT.EQ.6) WRITE(6,2015)
502:         WRITE(6,2000)((AVGTK(J,KT),J=1,12),(TKLCUM(J1,KT),J1=1,12),
503:         * (TKPCT(JJ,KT),JJ=1,12))
504:         WRITE(6,2001)((AVGAP(J,KT),J=1,12),(APLCUM(J1,KT),J1=1,12),
505:         * (APPCT(JJ,KT),JJ=1,12))
506:         WRITE(6,2002)((AVGHE(J,KT),J=1,5),(HELCUM(J1,KT),J1=1,5),
507:         * (HEPCT(JJ,KT),JJ=1,5))
508:         WRITE(6,2003)((AVGAT(J,KT),J=1,12),(ATLCUM(J1,KT),J1=1,12),
509:         * (ATPCT(JJ,KT),JJ=1,12))
510: 191 CONTINUE
511:     END

```

Figure III.15.3 (Cont)

CEN REPORT GENERATOR  
72KORC601-RUNSEC/KORC607-06 SC 00MAY80 UNCLASSIFIED

BLUE

COMBAT UNIT SUMMARY

ALL UNITS AT END OF DIVISION CYCLE 16

THEATER RESOURCES	COMBAT UNIT STATUS AFTER RESUPPLY				COMBAT UNIT SUPPORT BEFORE RESUPPLY				TOTAL UNIT LOSSES				CUMULATIVE COMBAT UNIT LOSSES			
	AUTH-ORIG	ON HAND	PERCENT	PERCENT	AVAILABLE	REQUIRED	PERCENT	PERCENT	LOSSES	LOSSES	LOSSES	LOSSES	TEMP	PERM	TEMP	PERM
PERSNL	394244.9	247784.2	61.3	51.3	8543.2	140878.0	5.3	112289.6	26712.2	250235.9	1763.2	870.8	20358.9			
1	39914.5	18919.5	47.4	47.4	43.2	21030.9	.2	6036.1	2217.0	19231.1	146.0	72.0	21607.0			
2	354230.4	222864.0	62.9	62.9	8500.0	139806.9	6.1	106373.6	24444.5	235004.0	1617.2	798.0	241914.0			
POL	1126757.0	1126423.0	100.0	100.0	0.0	0.0	0.0	43711.4	0.0	83164.5	0.0	1704.0	84070.6			
1	110293.0	110249.4	100.0	100.0	105051.7	471.7	999.9	3375.4	0.0	6011.5	0.0	279.5	7120.9			
2	1024464.0	1026143.6	100.0	100.0	0.0	5990.7	999.9	40336.0	0.0	76323.1	0.0	1424.6	77710.6			
ARMG	151060.7	151774.0	100.0	100.0	0.0	1497.6	999.9	1032.1	0.0	10533.6	0.0	51.6	10505.1			
1	11975.9	11978.5	100.0	100.0	0.0	105.3	999.9	799.2	0.0	6536.2	0.0	9.5	1545.7			
2	139082.8	139823.5	100.0	100.0	462150.7	1392.4	999.9	952.9	0.0	16977.4	0.0	42.1	17030.6			
TKNS 1	100.0	107.5	99.6	99.6	690.9	2.3	999.9	6.4	1.4	.7	4.0	.2	6.4			
TKNS 3	9.0	7.7	85.2	85.2	0.0	1.3	0.0	1.3	.9	.2	.2	.0	1.3			
TKNS 5	100.0	179.0	99.5	99.5	34.5	4.0	767.4	29.2	17.0	22.0	21.3	4.9	60.1			
TKNS 6	1292.0	920.4	71.0	71.0	32.9	391.0	0.4	293.4	228.5	200.7	130.6	22.2	450.0			
TKNSUM	1589.0	1284.6	77.1	77.1	760.3	400.2	192.0	330.3	239.7	322.5	164.2	27.4	783.0			
APCS 1	24.0	23.0	95.2	95.2	520.7	.9	999.9	4.2	2.9	.6	.6	.1	4.2			
APCS 2	167.0	104.1	99.4	99.4	305.7	4.5	999.9	31.0	15.1	1.9	4.2	.6	21.0			
APCS 3	42.0	41.0	99.6	99.6	211.0	.9	999.9	5.0	2.9	.8	1.1	.2	5.0			
APCS 4	21.0	20.0	99.5	99.5	129.9	.7	999.9	2.0	1.6	.2	.8	.1	2.0			

Figure III.15.4

BLUE

COMBAT UNIT SUMMARY

ALL UNITS AT END OF DIVISION CYCLE 16

TREATY RESOURCES	COMBAT UNIT STATUS AFTER RESUPPLY				COMBAT UNIT SUPPORT BEFORE RESUPPLY				TOTAL UNIT LOSSES		COMBAT		NONCOMBAT		CUMULATIVE COMBAT UNIT LOSSES	
	AUTM ORIGD	ON HAND	PER- CENT	PER- CENT	AVAILABLE REQUIRED	PER- CENT	PER- CENT	PER- CENT	TEMP	PERM	TEMP	PERM	TEMP	PERM	TEMP	PERM
APCS 5	104.0	70.3	67.6		6.0	40.2	16.0		40.0		64.0	20.1	6.7	1.2	100.0	
APCS 6	759.0	562.1	74.6		24.9	216.0	11.5		152.1		257.0	81.2	40.2	11.1	410.3	
APCS 7	150.0	144.8	96.5		156.2	13.5	99.9		72.9		80.2	75.3	14.5	2.6	172.7	
APCSUM	1262.0	1029.0	81.6		1373.1	277.5	99.9		290.0		924.0	180.2	80.2	15.9	710.0	
(CONTINUED FROM PRECEDING PAGE)																
HELO 1	40.0	43.7	91.0		1.6	6.0	27.2		17.3		3.9	10.2	3.0	.3	17.3	
HELO 3	54.0	43.0	81.1		3.1	17.4	12.2		54.3		26.5	50.7	11.0	1.9	90.1	
HELO 4	74.0	49.0	67.3		.0	24.2	.0		24.2		6.4	3.2	4.1	.5	29.2	
HELOSUM	176.0	137.3	78.0		3.0	47.5	7.9		95.8		36.8	62.1	18.1	2.7	130.6	
AT/M 1	45.0	45.0	100.0		9999.0	.0	.0		.0		.0	.0	.0	.0	.0	
AT/M 3	147.0	146.1	99.7		9995.1	9.4	99.9		23.4		.0	23.4	.0	.0	23.4	
AT/M 4	96.0	92.4	96.3		81.7	17.0	95.1		44.1		.0	44.1	.0	.0	44.1	
AT/M 5	2377.0	2105.0	92.9		52.6	224.7	23.4		461.2		.0	1014.0	.0	.0	1014.0	
AT/M 6	2234.0	1303.5	58.3		.0	920.5	.0		339.7		.0	920.5	.0	.0	920.5	
AT/M 7	115.0	114.1	99.2		0702.7	4.6	99.9		25.7		.0	60.0	.0	.0	60.0	
AT/M 8	350.0	224.0	64.2		4.0	129.2	3.1		59.2		.0	169.2	.0	.0	169.2	
AT/M 9	334.0	334.0	100.0		1604.0	.0	.0		.0		.0	.0	.0	.0	.0	
AT/M10	1995.0	1995.0	100.0		910.0	.0	.0		.0		.0	.0	.0	.0	.0	
AT/M11	66.0	66.0	100.0		312.0	.0	.0		.0		.0	.0	.0	.0	.0	

Figure III.15.4 (Cont)

UNCLASSIFIED--EXAMPLE OF CEMLOSSI OUTPUT DATA FROM UTILITY CEM/DATA

1: NUMBER OF 30-DAY BLOCKS OF DATA = 6

AVERAGE 15-DAY LOSS RATES \*\*D-1 TO D-15\*\*

	1	2	3	4	5	6	7	8	9	10	11	12
EQUIP TYPE	1	2	3	4	5	6	7	8	9	10	11	12
AVG TK AUTH	1369.40	1838.00	.00	324.00	.00	.00	.00	.00	.00	.00	.00	.00
PCT TK LOSS	.32	.07	.00	.05	12.39	58.56	4.65	348.47	135.66	58.95	7.15	27.08
AVG APC AUTH	1297.53	2519.07	1244.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
PCT APC LOSS	.12	.11	.18	13.17	1.35	7.97	13.10	126.64	36.76	225.05	217.27	9.53
AVG MEL AJTH	235.40	.00	.00	2.60	180.00							
PCT MEL LOSS	.36	3.32	55.97	.75	.35							
AVG ATM AUTH	492.73	572.87	2241.53	.00	.00	.00	.00	.00	.00	.00	.00	.00
PCT ATM LOSS	.00	.00	.03	.00	92.62	147.30	326.03	25.58	.00	.00	.00	.00

AVERAGE 30-DAY LOSS RATES \*\*D-1 TO D-30\*\*

	1	2	3	4	5	6	7	8	9	10	11	12
EQUIP TYPE	1	2	3	4	5	6	7	8	9	10 <td>11</td> <td>12</td>	11	12
AVG TK AUTH	1461.93	2058.00	43.20	324.00	.00	.00	.00	.00	.00	.00	.00	.00
PCT TK LOSS	.28	.07	.61	.10	5.42	9.98	34.50	102.00	70.93	26.13	47.82	13.72
AVG APC AUTH	1486.60	2730.67	1628.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
PCT APC LOSS	.17	.12	.26	7.53	2.06	8.19	89.24	58.41	14.75	149.91	110.82	77.73
AVG MEL AUTH	411.40	.00	.00	14.20	204.00							
PCT MEL LOSS	.30	12.65	8.70	.42	.21							
AVG ATM AUTH	841.53	634.73	3251.13	129.60	.00	.00	.00	.00	.00	.00	.00	.00
PCT ATM LOSS	.00	.00	.09	.06	38.72	54.12	105.55	11.38	.00	.00	.00	.00

Fig. 15.5

[illegible]

AVERAGE 30-DAY LOSS RATES 00-1 TO 0-6000

[illegible]

AVERAGE 30-DAY LOSS RATES \*\*D-1 TO D-90\*\*

[illegible]

349

UNCLASSIFIED\*\*EXAMPLE OF CEM LOSS1 OUTPUT DATA FROM UTILITY CEM/DATA

115:	PCT	ATM	LOSS	.00	.00	.14	.07	.00	.00	.00	.00	.00	.00	.00	.00
116:															
117:															
118:															
119:															
120:															
121:															
122:															
123:															
124:	EQUIP	TYPE	1	2	3	4	5	6	7	8	9	10	11	12	
125:	AVG	TK	AUTH	4768.00	2058.00	54.00	324.00	.00	.00	.00	.00	.00	.00	.00	.00
126:	PCT	TK	LOSS	.19	.04	.00	.05	.00	.00	.00	.00	.00	.00	.00	.00
127:															
128:	AVG	APC	AUTH	2190.00	5526.00	3587.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
129:	PCT	APC	LOSS	.13	.07	.16	.00	.00	.00	.00	.00	.00	.00	.00	.00
130:															
131:															
132:	AVG	MEL	AUTH	797.00	.00	.00	29.50	240.00							
133:	PCT	MEL	LOSS	.04	.00	.00	.16	.05							
134:															
135:															
136:	AVG	ATM	AUTH	2174.00	1204.00	7306.00	162.00	.00	.00	.00	.00	.00	.00	.00	.00
137:	PCT	ATM	LOSS	.00	.00	.06	.01	.00	.00	.00	.00	.00	.00	.00	.00
138:															
139:															
140:															
141:															
142:															
143:															
144:															
145:															
146:															
147:	EQUIP	TYPE	1	2	3	4	5	6	7	8	9	10	11	12	
148:	AVG	TK	AUTH	4768.00	2058.00	54.00	324.00	.00	.00	.00	.00	.00	.00	.00	.00
149:	PCT	TK	LOSS	.24	.04	.37	.05	.00	.00	.00	.00	.00	.00	.00	.00
150:															
151:	AVG	APC	AUTH	2190.00	5526.00	3587.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
152:	PCT	APC	LOSS	.13	.09	.24	.00	.00	.00	.00	.00	.00	.00	.00	.00
153:															
154:															
155:	AVG	MEL	AUTH	797.00	.00	.00	29.50	240.00							
156:	PCT	MEL	LOSS	.04	.00	.00	.14	.03							
157:															
158:															
159:	AVG	ATM	AUTH	2174.00	1204.00	7306.00	162.00	.00	.00	.00	.00	.00	.00	.00	.00
160:	PCT	ATM	LOSS	.00	.00	.05	.01	.00	.00	.00	.00	.00	.00	.00	.00
161:															
162:															
163:															
164:															
165:															
166:															
167:															
168:															
169:	EQUIP	TYPE	1	2	3	4	5	6	7	8	9	10	11	12	
170:	AVG	TK	AUTH	4768.00	2058.00	54.00	324.00	.00	.00	.00	.00	.00	.00	.00	.00
171:															

Figure III.15.5 (Cont)

172:	PCT TK	LOSS	.22	.03	.62	.04	.00	.00	.00	.00	.00
173:											.0C
174:											
175:	AVG APC AUTH	2190.00	5526.00	3587.00	.00	.00	.00	.00	.00	.00	.0C
176:	PCT APC LOSS	.11	.08	.21	.00	.00	.00	.00	.00	.00	.00
177:											.00
178:											
179:	AVG HEL AUTH	797.00	.00	.00	29.50	.11	.03				
180:	PCT HEL LOSS	.03	.00	.00							
181:											
182:											
183:	AVG ATM AUTH	2174.00	1204.00	7306.00	162.00	.00	.00	.00	.00	.00	.0C
184:	PCT ATM LOSS	.00	.00	.04	.00	.00	.00	.00	.00	.00	.0C

30-DAY LOSS RATE \*\* D-1 TO D-30\*\*

196:	AUG TANK	AUTH	1425.67	1948.00	21.60	324.00	.00	.00	.00	.00	.00	.00
197:	TOT TANK	LOSS	855.66	262.49	26.20	47.49	17.81	68.55	39.15	450.47	206.59	85.08
198:	CT TANK	LOSS	.60	.13	1.21	.15	.00	.00	.00	.00	.00	.00
199:												
200:												
201:	AUG APC	AUTH	1392.07	2624.87	1436.00	.00	.00	.00	.00	.00	.00	.00
202:	TOT APC	LOSS	407.43	587.07	648.69	20.70	3.41	16.25	102.34	185.05	51.51	374.96
203:	PCT APC	LOSS	.29	.22	.45	.00	.00	.00	.00	.00	.00	.00
204:												
205:												
206:	AUG HELO	AUTH	323.40	.00	.00	13.40	192.00					
207:	TOT HELO	LOSS	209.77	15.96	64.67	15.40	113.53					
208:	PCT HELO	LOSS	.65	.00	.00	1.15	.59					

[illegible]

351

[illegible]

352

30-DAY LOSS RATE ** D-151 TO D-180**										
296:	AVG TANK AUTH	4768.00	2058.00	54.00	324.00	.00	27.68	.00	.00	.00
297:	TOT TANK LOSS	1409.55	70.78	39.86	14.20	.04	.00	.00	.00	.00
298:	CT TANK LOSS	.30	.03	.74	.04	.00	.00	.00	.00	.00
299:	AVG APC AUTH	2190.00	5526.00	3587.00	.00	.00	10.72	.00	.00	.00
300:	TOT APC LOSS	314.70	577.90	1185.64	17.20	2.57	.00	.00	.00	.00
301:	PCT APC LOSS	.14	.10	.33	.00	.00	.00	.00	.00	.00
302:	AVG HELO AUTH	797.00	.00	.00	29.50	240.00	.00	.00	.00	.00
303:	TOT HELO LOSS	23.18	5.86	4.46	3.67	3.33	.00	.00	.00	.00
304:	PCT HELO LOSS	.03	.00	.00	.12	.01	.00	.00	.00	.00
305:	AVG ATM AUTH	2174.00	1204.00	7306.00	162.00	.00	.00	.00	.00	.00
306:	TOT ATM LOSS	.00	.00	316.86	.00	23.45	8.32	120.77	.00	.00
307:	PCT ATM LOSS	.00	.00	.04	.00	.00	.00	.00	.00	.00
308:	AVG TANK AUTH	4768.00	2058.00	54.00	324.00	.00	.00	.00	.00	.00
309:	TOT TANK LOSS	801.24	21.05	92.65	9.18	3.85	25.02	6.26	.00	.00
310:	CT TANK LOSS	.17	.01	1.72	.03	.00	.00	.00	.00	.00
311:	AVG APC AUTH	2190.00	5526.00	3587.00	.00	.00	.00	.00	.00	.00
312:	TOT APC LOSS	144.03	285.13	522.34	13.73	2.13	11.08	36.48	.00	.00
313:	PCT APC LOSS	.07	.05	.15	.00	.00	.00	.00	.00	.00
314:	AVG HELO AUTH	797.00	.00	.00	29.50	240.00	.00	.00	.00	.00
315:	TOT HELO LOSS	11.34	1.12	3.20	1.37	1.95	.00	.00	.00	.00
316:	PCT HELO LOSS	.01	.00	.00	.05	.01	.00	.00	.00	.00
317:	AVG ATM AUTH	2174.00	1204.00	7306.00	162.00	.00	.00	.00	.00	.00
318:	TOT ATM LOSS	.00	.00	45.87	.14	12.28	5.55	63.01	.00	.00
319:	PCT ATM LOSS	.00	.00	.01	.00	.00	.00	.00	.00	.00
320:	AVG TANK AUTH	4768.00	2058.00	54.00	324.00	.00	.00	.00	.00	.00
321:	TOT TANK LOSS	801.24	21.05	92.65	9.18	3.85	25.02	6.26	.00	.00
322:	CT TANK LOSS	.17	.01	1.72	.03	.00	.00	.00	.00	.00
323:	AVG APC AUTH	2190.00	5526.00	3587.00	.00	.00	.00	.00	.00	.00
324:	TOT APC LOSS	144.03	285.13	522.34	13.73	2.13	11.08	36.48	.00	.00
325:	PCT APC LOSS	.07	.05	.15	.00	.00	.00	.00	.00	.00
326:	AVG HELO AUTH	797.00	.00	.00	29.50	240.00	.00	.00	.00	.00
327:	TOT HELO LOSS	11.34	1.12	3.20	1.37	1.95	.00	.00	.00	.00
328:	PCT HELO LOSS	.01	.00	.00	.05	.01	.00	.00	.00	.00
329:	AVG ATM AUTH	2174.00	1204.00	7306.00	162.00	.00	.00	.00	.00	.00
330:	TOT ATM LOSS	.00	.00	45.87	.14	12.28	5.55	63.01	.00	.00
331:	PCT ATM LOSS	.00	.00	.01	.00	.00	.00	.00	.00	.00
332:	AVG TANK AUTH	4768.00	2058.00	54.00	324.00	.00	.00	.00	.00	.00
333:	TOT TANK LOSS	80								

353

THIS PAGE IS INTENTIONALLY LEFT BLANK

## CHAPTER 16

### Utility - CEM/LOSSES

**16.1 DESCRIPTION OF PROCESSING:** In addition to logical tested read and write statements, the program performs computations on the data to produce the rates. The program has calls to one subroutine.

**16.1.1 PURPOSE/FUNCTIONS:** The purpose of this module is to produce the CEM/LOSSES file which will be used later as input to the CONTROL/COMPILER utility. This utility will use as its primary input the 82CEMLOSS1 file which was produced by the preceding utility, CEM/DATA. Further, it also requires the user to provide a list of specific CEM weapon control numbers identifying equipment models for which CEM loss rates are needed. Using this information the utility will read the 82CEMLOSS1 file and screen out unneeded data such as the authorization levels, and produce a summary of the CEM/LOSS rates for all models of equipment in the study, further it will select the loss rates for those models identified by the user in the runstream and write a summary line of these loss rates and express them as percentages.

**16.1.2 PROGRAM INPUT/OUTPUT STRUCTURE:** The overall structure of the utility is pictured in FIGURE III.16.1. The logic flow followed by the utility is pictured in FIGURE III.16.2. The source code of the utility is listed in Figure III.16.4.

**16.1.2.A INPUT DATA AND DATA BASE:** The major input to this utility is the 82CEMLOSS1 file which was produced by the previous utility CEM/DATA. This file summarized information received from the CEM LOG report as to the authorized number and loss rates for up to 12 individual models of four major types of combat equipment, i.e., tanks, APC's, helicopters and ATM's. It accumulates this information by the various time periods of the study.

Figure III.6.5 presents an example of the data found in the file, since the file is produced in a report-like format which is self-explanatory.

Also used as input to this utility is a list of two character CEM Weapon Numbers which controls the output of loss rates which is produced. This list is entered into the runstream of the utility. Table III.16.15 lists examples of these numbers and provides an explanation. These numbers change from study to study and current list must be coordinated with CEM Analyst. Refer to Volume I for the program runstream.

**16.1.2.B OUTPUT DATA AND DATA FILES:** The CEM/LOSSES utility produces the CEM/LOSSES file as its single output. This file will be used as one of the input files to a subsequent utility, CONTROL/COMPILER. The file itself is quite similar in appearance to the 82CEMLOSS1 file in that it summarizes over a number of time periods information concerning loss rate up to 12 models of four major types of combat equipment, i.e., tanks, APC's, helicopters and ATM's. Further it is formatted in such a manner that it is quite readable by itself. But the file is unique in that it does not contain any authorization levels for the equipment, only loss rates. Further the last line or record of each time period is

entitled "RO9CEM" or "R10CEM" denotes the loss rates for those Weapon Systems explicitly entered by the user in the runstream. Figure III.16.6 presents an example of this output file.

#### 16.1.2.C DATA ELEMENT DICTIONARY:

The following section identifies and defines all variables used in this utility.

<u>NAME</u>	<u>DEFINITION</u>								
IPOSIT	A single dimension integer variable with 22 occurrences. Each occurrence holds the 2 character identification number of the up to 22 major items of equipment that are being analyzed. If more than 22 items are being analyzed the sized of this dimension must be increased and the utility recompiled. This data is read from the job stream using logical unit 5.								
TEMP	This is a single dimension variable which has 12 occurrences. The array is used as a dummy to read past data in the **CEMLOSSES1 input file which is not needed by the utility.								
IPERID	This is a 2 character integer variable which will contain the number of days used as the basis for calculating the loss rates. For example if IPERID=30, this indicates that the following loss data will reflect average 30 day loss rates, regardless of the number of days in the particular period. This data is read from the **CEMLOSSES1 input file.								
IDAY1	This is a 2 character integer variable which indicates the start day of the particular time period. This data is read from the **CEMLOSSES1 input file.								
IDAY2	This is a 2 character integer variable which indicates the ending day of a particular time period. This data is read from the **CEMLOSSES1 input file.								
PNUM	<p>A single dimension array of 41 occurrences. Entries in this array will specify percentage loss rates experienced by each model of major equipment type being played. The 41 occurrences are allocated to the 4 major equipment types using the following scheme:</p> <table><tr><td>PNUM 1-12</td><td>Tanks</td></tr><tr><td>PNUM 13-24</td><td>APC's</td></tr><tr><td>PNUM 25-29</td><td>Helicopters</td></tr><tr><td>PNUM 30-41</td><td>ATM's</td></tr></table> <p>This data will be read from the input file **CEMLOSSES1. In order to express the input data as</p>	PNUM 1-12	Tanks	PNUM 13-24	APC's	PNUM 25-29	Helicopters	PNUM 30-41	ATM's
PNUM 1-12	Tanks								
PNUM 13-24	APC's								
PNUM 25-29	Helicopters								
PNUM 30-41	ATM's								

percentages, they are multiplied by 100. Further, if the loss data are for periods where rates are not based upon 15 days (i.e., IPERID.NE.15), then the PNUM values are doubled.

**ISTART**

An integer variable calculated in the utility which indicates the starting point within the PNUM array where loss data being read in should start.

**ISTOP**

An integer variable calculated in the utility which indicates the ending point within the PNUM array where loss data being read in should stop.

**OUTPNT**

A single dimension array of 22 occurrences. The entries in this array will be loss percentage rate values found in the PNUM array, which loss percentage rates from PNUM are to be assigned to this array is controlled by the entries in the IPOSIT array. These entries in the IPOSIT array denote the occurrence in the PNUM array (i.e., the CEM weapon control numbers which are being played in this study). That is to be assigned to the particular occurrence of OUTPNT. For example if IPOSIT=25 it would indicate that the loss percentage rate from the first helicopter model is being assigned to the current occurrence of OUTPNT. The values of OUTPNT are also the ones printed OUI in the CEM/LOSSES output file.

**ITEMP**

A temporary integer variable which is used to hold the value of the current occurrence of the IPOSIT array. This variable is subsequently used as a subscript in the PNUM array to identify the proper occurrence of PNUM to be assigned to OUTPNT.

### CEM Weapon Numbers List (Example)

<u>Number</u>	<u>Example</u>
01	M1 TANK
02	M60A1 TANK
03	M551 Assault vehicle
04	M48A7 TANK
13	IFV
15	M113A1 APC
25	AAH
29	AH-15
32	DRAGON

Table III.16.1

**16.2 OPERATING ENVIRONMENT:** This program is implemented on the EXECUTIVE-8 operating system.

**16.2.1 SUPPORT SOFTWARE:** This utility requires the FORTRAN IV compiler and the UNIVAC 1100/02 system facilities.

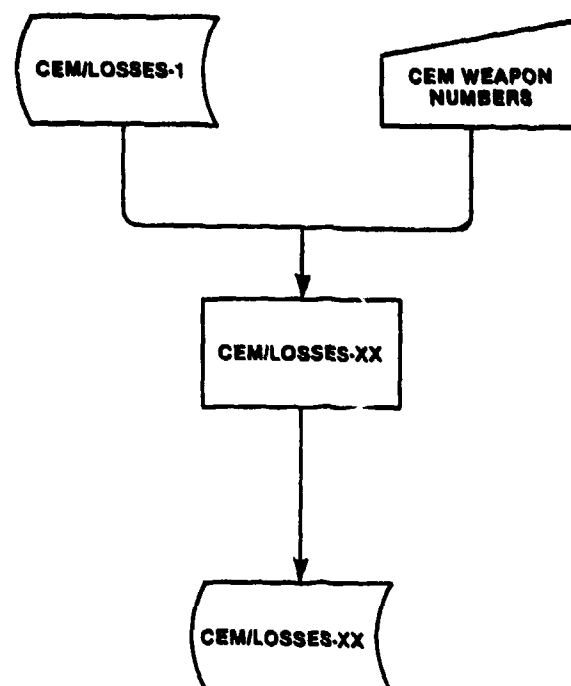
**16.2.2 I/O DEVICES:** The utility uses input files which reside on disk. It produces an output file which will also reside on disk.

**16.3 MAINTENANCE PROCEDURES:** This program is maintained by the MPP analyst.

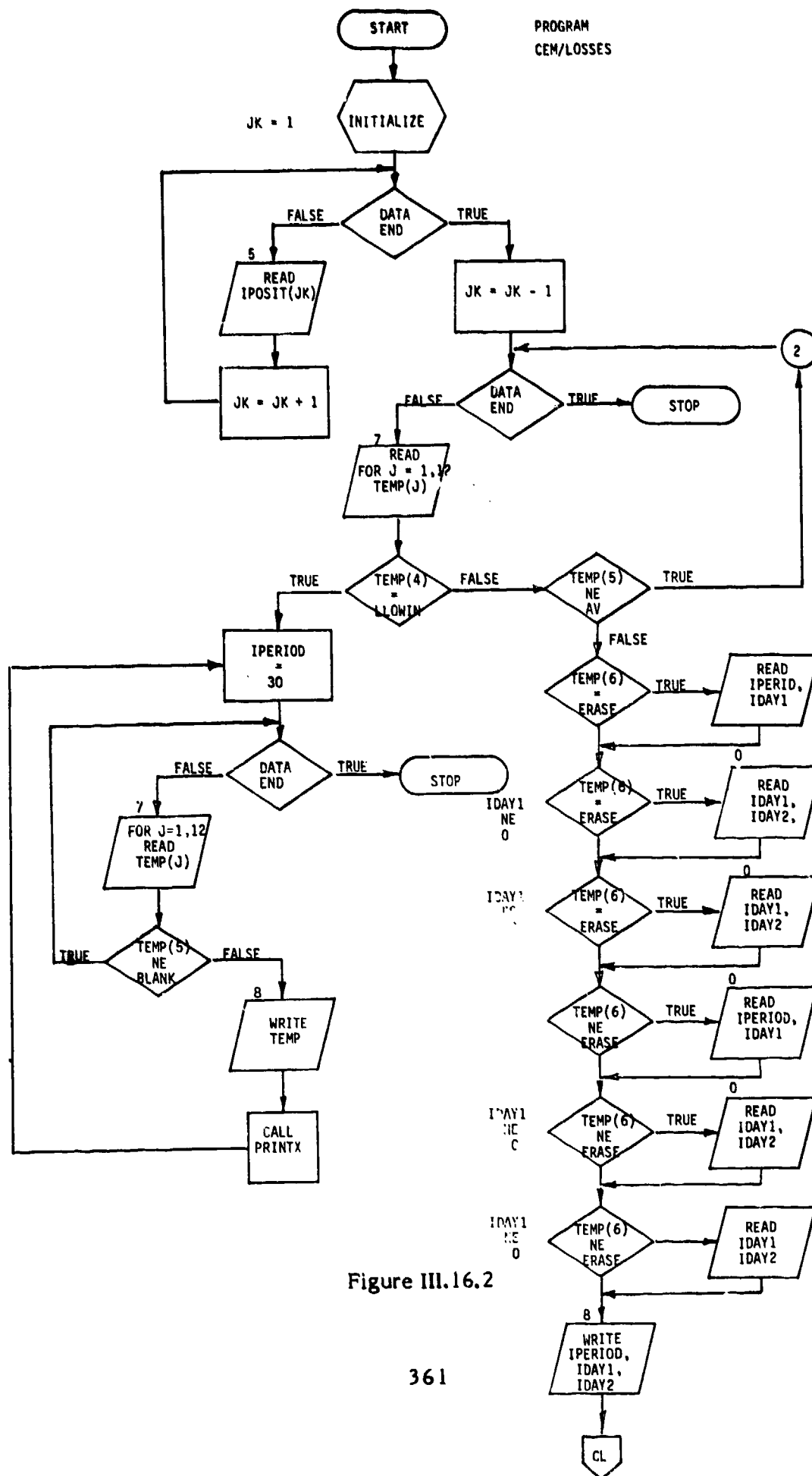
**16.3.1 PROGRAMMING CONVENTIONS:** The utility uses one subroutine in its structure. Standard FORTRAN conventions are followed. The subroutine is maintained as source code in the same element as the driving program.

**16.3.2 INTERNAL ERROR ROUTINES:** There are no explicit error handling written into the utility. As a result the only error detection facilities will be provided by the system.

**CEM/LOSSES STRUCTURE**



**Figure III.16.1**



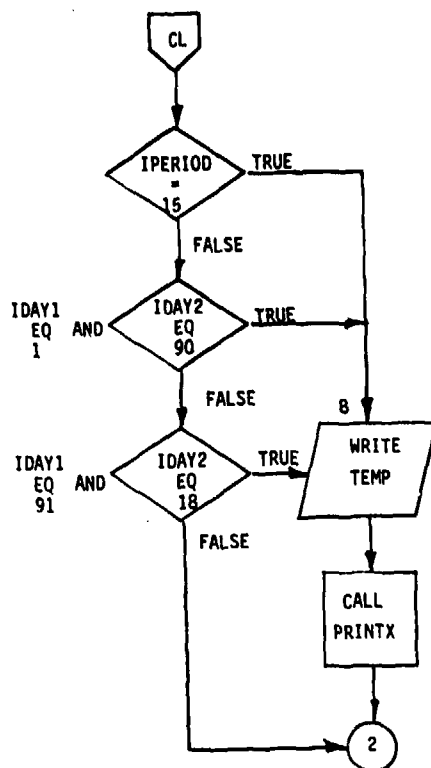


Figure III.16.2 (Cont)

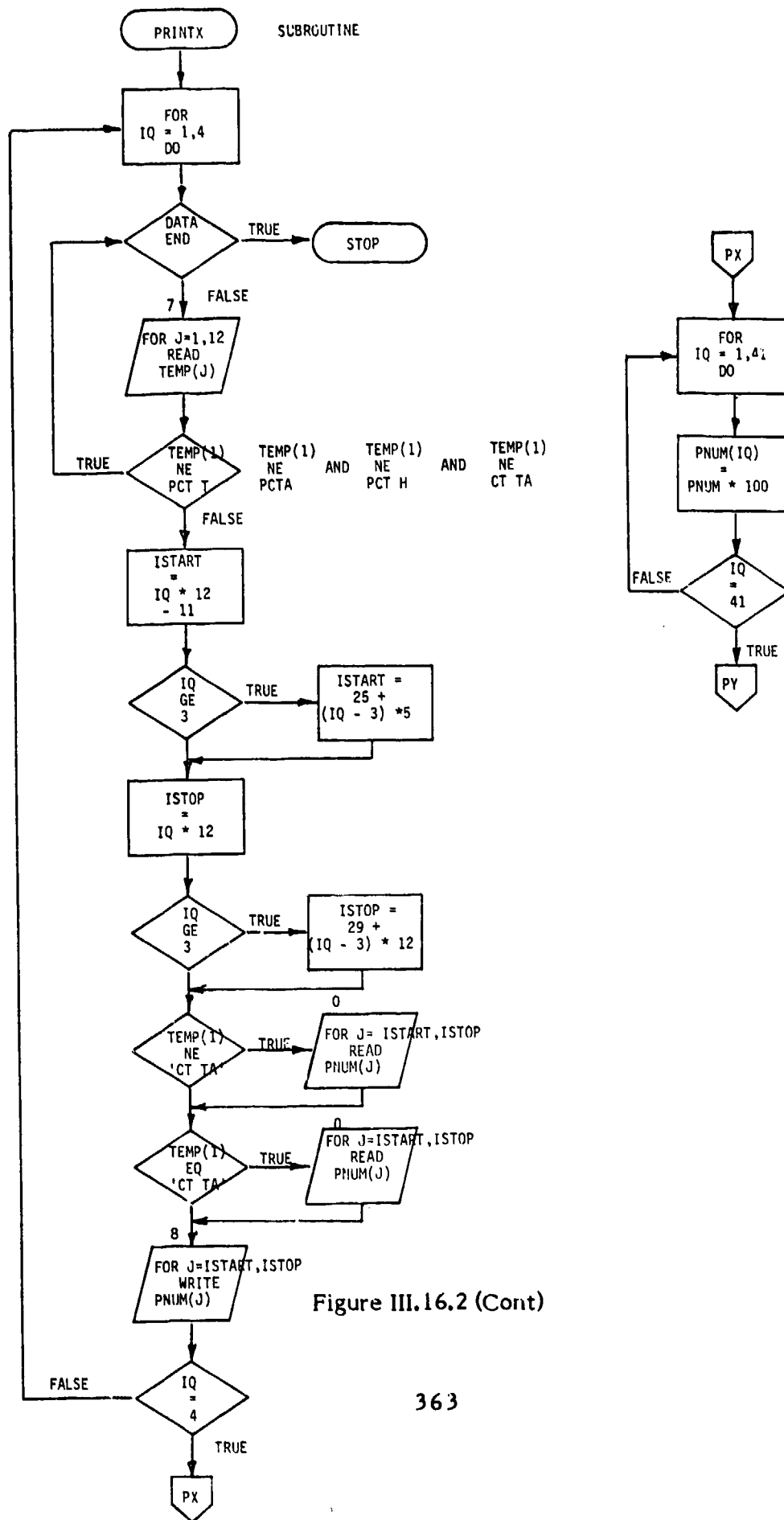


Figure III.16.2 (Cont)

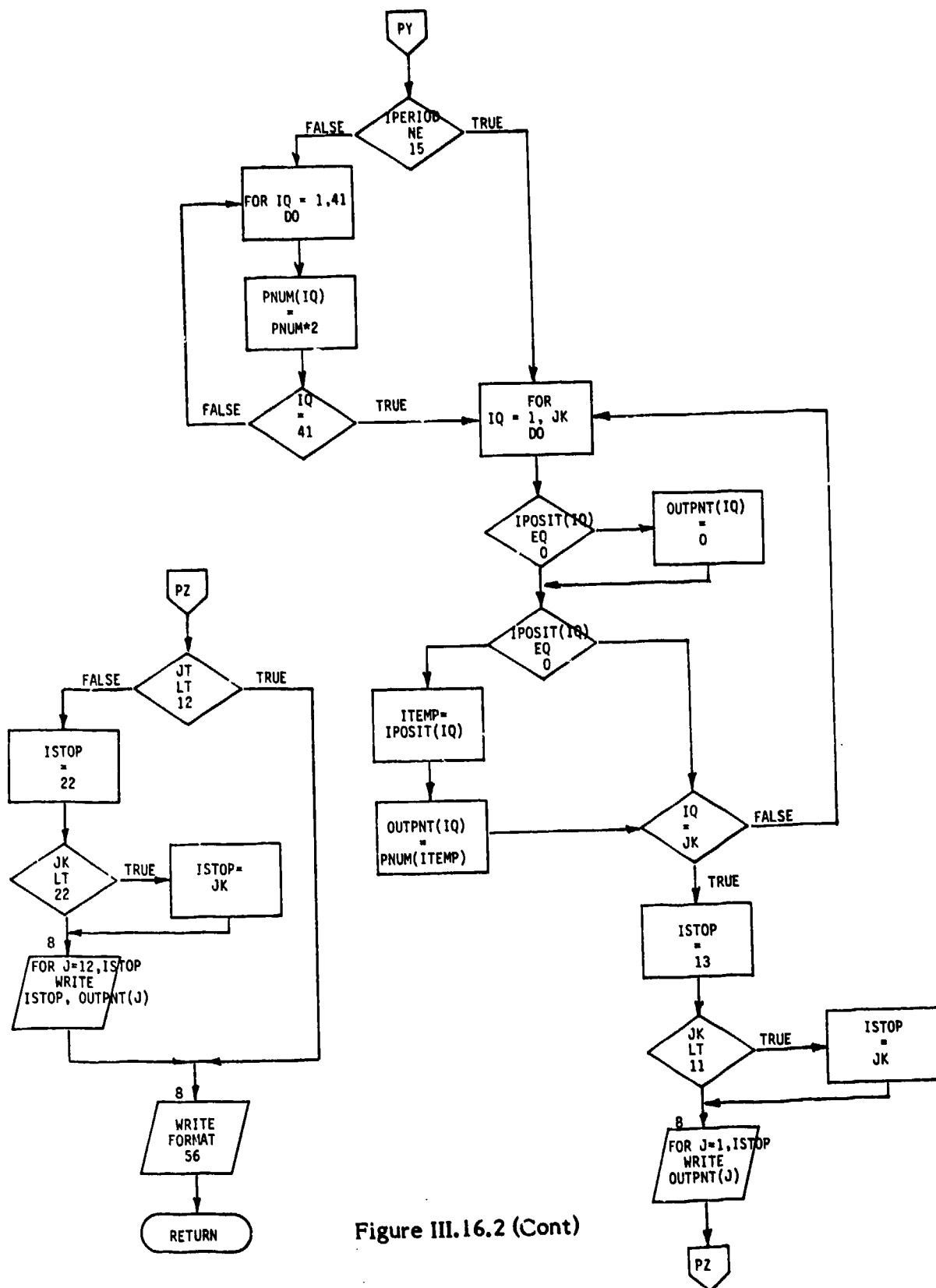


Figure III.16.2 (Cont)

```

1:C*****
2:C*****
3:C*****THIS PROGRAM IS WRITTEN IN FORTRAN IV.
4:C*****
5:C*****
6:C*****PURPOSE OF PROGRAM:
7:C*****
8:C*****THIS UTILITY CREATES CEM LOSS RATES FOR EACH OF THE SEVEN
9:C*****TIME PERIODS ON ONLY THOSE MIE OF INTEREST TO THE WARF STUDY.
10:C*****THIS PROGRAM USES AS INPUT:
11:C***** A. THE OUTPUT OF 82XQT.CEM/DATA (DATA FILE 82CEMLOSS1).
12:C***** B. FROM THE RUN STREAM THE CEM WEAPON CONTROL NUMBERS.
13:C*****THERE IS A MAXIMUM OF 22 POSITIONS OF OUTPUT FOR THIS
14:C*****UTILITY, THUS LIMITING THE NUMBER OF WEAPON SYSTEMS
15:C*****UPON WHICH LOSS RATES CAN BE COMPUTED TO 22. THIS PROBLEM CAN
16:C*****BE EASILY SOLVED BY CHANGING THE APPROPRIATE DIMENSION, READ
17:C*****AND WRITE STATEMENTS.
18:C*****THE LOSS RATES COMPUTED HERE ON EACH WARF MIE PLAYED IN
19:C*****CEM WILL BE USE AS INPUT BY THE UTILITY "82XQT.CONTROL/
20:C*****COMPILER".
21:C*****
22:C*****
23:      DIMENSION IPOSIT(22),TEMP(12),PNJH(41),OUTPNT(22)
24:      JK=1
25:1      READ(5,10,END=20)IPOSIT(JK)
26:10     FORMAT(I2)
27:      JK=JK+1
28:      GO TO 1
29:20     JK=JK-1
30:C*****
31:C*****
32:2      READ(7,25,END=999)(TEMP(J),J=1,12)
33:25     FORMAT(12A6)
34:      IF(TEMP(4).EQ.'LOWIN') GO TO 3
35:      IF(TEMP(5).NE.' AV')GO TO 2
36:      IF(TEMP(6).EQ.'EPAGE ')READ(10,26)IPERID,1DAY1
37:26     FORMAT(36X,I2,21X,I1)
38:      IF((TEMP(6).EQ.'EPAGE ').AND.(1DAY1.NE.0))READ(10,18)1DAY1,1DAY2
39:18     FORMAT(58X,I2,6X,I2)
40:      IF((TEMP(6).EQ.'EPAGE ').AND.(1DAY1.EQ.0))READ(10,19)1DAY1,1DAY2
41:19     FORMAT(55X,I1,6X,I2)
42:      IF(TEMP(6).NE.'EPAGE ')READ(10,27)IPERID,1DAY1
43:27     FORMAT(32X,I2,21X,I1)
44:      IF((TEMP(6).EQ.'EPAGE ').AND.(1DAY1.NE.0))READ(10,16)1DAY1,1DAY2
45:16     FORMAT(54X,I2,6X,I2)
46:      IF((TEMP(6).NE.'EPAGE ').AND.(1DAY1.EQ.0))READ(10,17)1DAY1,1DAY2
47:17     FORMAT(54X,I1,6X,I2)
48:      WRITE(10,568)IPERID,1DAY1,1DAY2
49:568     FORMAT(3I4)
50:      IF(IPERID.EQ.15)GO TO 28
51:      IF((1DAY1.EQ.1).AND.(1DAY2.EQ.9))GO TO 28
52:      IF((1DAY1.EQ.9).AND.(1DAY2.EQ.16))GO TO 28
53:      GO TO 2
54:28     WRITE(18,29)TEMP
55:29     FORMAT(12A6)
56:      CALL PRINTX
57:      GO TO 2
58:C*****
59:C*****
60:3      IPERID=30
61:4      READ(7,25,END=999)(TEMP(J),J=1,12)
62:      IF(TEMP(5).NE.' 30')GO TO 4
63:      WRITE(18,29)TEMP
64:      CALL PRINTX
65:      GO TO 3
66:C*****
67:C*****
68:999     STOP

```

Figure III.16.3

```

64:      SUBROUTINE PRINTX
65:      DO 35 IQ=1,4
71:5     READ(7,32,END=998)(TEMP(J),J=1,12)
72:32    FORMAT(12A6)
73:C*****WRITE(8,887)TEMP(1)
74:887   FORMAT('**',A6,'**')
75:      IF((TEMP(1).NE.' PCT T').AND.(TEMP(1).NE.' PCT A'))
76:      2.AND.(TEMP(1).NE.' PCT H').AND.(TEMP(1).NE.' CT TA'))
77:      3GO TO 5
78:C*****WRITE(8,886)
79:886   FORMAT(21X,'REACHED LINE 73')
80:      ISTART=IQ * 12 - 11
81:      IF(IQ.GE.3)ISTART= 25 + (IQ-3)*5
82:      ISTOP=IQ*12
83:      IF(IQ.GE.3)ISTOP= 29 + (IQ-3)*12
84:      IF(TEMP(1).NE.' CT TA')READ(C,30)(PNUM(J),J=ISTART,ISTOP)
85:30     FORMAT(14X,12F5.2)
86:      IF(TEMP(1).EQ.' CT TA')READ(D,31)(PNUM(J),J=ISTART,ISTOP)
87:31     FORMAT(13X,12F8.2)
88:      WRITE(8,885)(PNUM(J),J=ISTART,ISTOP)
89:885   FORMAT(5X,12F8.2)
90:35    CONTINUE
91:C*****
92:C*****
93:      DO 39 IO=1,41
94:      PNUM(IO)=PNUM(IO)*100
95:39    CONTINUE
96:C*****
97:C*****
98:      IF(IPENIO.NE.15)GO TO 40
99:C*****
100:C*****
101:      DO 35 IO=1,41
102:      PNUM(IO)=PNUM(IO)*2
103:35    CONTINUE
104:C*****
105:C*****
106:40    DO 45 IQ=1,JK
107:      IF(IPOSIT(IQ).EQ.0) OUTPNT(IQ)=0
108:      IF(IPOSIT(IQ).EQ.0)GO TO 45
109:      ITEMP=IPOSIT(IQ)
110:      OUTPNT(IQ)=SUM(ITEMP)
111:45    CONTINUE
112:      ISTOP=13
113:      IF(JK.LT.11)ISTOP=JK
114:      WRITE(8,57)ISTOP,(OUTPNT(J),J=1,ISTOP)
115:50     FORMAT('NOVCEM  G1-',I2,13F7.1)
116:C*****
117:C*****
118:      IF(JK.LT.12)GO TO 55
119:      ISTOP=22
120:      IF(JK.LT.20)ISTOP=JK
121:      WRITE(8,51)ISTOP,(OUTPNT(J),J=12,ISTOP)
122:51     FORMAT('RICECM  I2-',I2,13F7.1)
123:55     WRITE(8,56)
124:56     FORMAT(1X,////)
125:      RETURN
126:998   STOP
127:      END

```

Figure III.16.4

## 1: NUMBER OF 30-DAY BLOCKS OF DATA = 6

367

**Figure III.16.5**

[illegible]

368

115:	PCT ATM LOSS	.00	.14	.07	.06	.00	.00	.00
116:								
117:								
118:								

Avg 30-day Loss Rates 00-91 to 0-12000	Sustaining Period
0.00%	00-91
0.00%	0-12000

	1	2	3	4	5	6	7	8	9	10	11	12
EQUIP TYPE												
Avg TK AUTH	\$768.00	2059.00	54.00	324.00	.00	.00	.00	.00	.00	.00	.00	.00
PCT TK LOSS	.19	.04	.00	.05	.00	.00	.00	.10	.00	.00	.00	.00

[illegible]

AVE HEL AUTH	.00	29.50	240.00
PCT HEL LOSS	.04	.00	.05

[illegible]

AVG 30-DAY LOSS RATES \*\*0-91 TO D-150\*\*

[illegible]

Avg APC Auth	2190.00	5526.00	3587.00	.00	.00	.00	.00	.00	.00
--------------	---------	---------	---------	-----	-----	-----	-----	-----	-----

AVG HEL AUTH	797.00	.00	29.50	240.00
8-7 HEL INCE	24	00	04	07

[illegible]

369



[illegible]

371

30-DAY LOSS RATE ** D-151 TO D-180**									
29C:									
29T:									
29B:	AVG TANK AUTH	4768.00	2059.00	54.00	324.00	.00	.00	.00	.00
29S:	TOT TANK LOSS	1409.95	70.78	39.36	14.20	3.20	27.68	37.29	32.73
29D:	CT TANK LOSS	.50	.03	.74	.04	.00	.00	.00	.00
29E:									
29F:									
29G:	AVG APC AUTH	2190.00	5524.00	3507.00	.00	.00	.00	.00	.00
29H:	TOT APC LOSS	314.70	577.90	1105.64	17.20	2.57	10.72	132.57	66.16
29I:	PCT APC LOSS	.14	.10	.33	.04	.00	.00	.00	.00
29J:									
29K:	AVG HELO AUTH	797.00	.00	.00	29.50	240.00	.00	.00	.00
29L:	TOT HELO LOSS	23.18	5.86	4.46	3.67	3.33	.00	.00	.00
29M:	PCT HELO LOSS	.03	.00	.00	.12	.01	.00	.00	.00
29N:									
29O:	AVG ATM AUTH	2174.00	1204.00	7106.00	162.00	.00	.00	.00	.00
29P:	TOT ATM LOSS	.00	.00	316.96	.00	23.45	9.12	120.77	8.05
29Q:	PCT ATM LOSS	.00	.00	.04	.00	.00	.00	.00	.00
29R:									
29S:	AVG TANK AUTH	4768.00	2058.00	54.00	324.00	.00	.00	.00	.00
29T:	TOT TANK LOSS	801.24	21.05	32.65	5.19	3.05	25.02	6.26	7.74
29U:	CT-TANK LOSS	.17	.01	1.72	.03	.00	.00	.00	.00
29V:									
29W:	AVG APC AUTH	2190.00	5526.00	3507.00	.00	.00	.00	.00	.00
29X:	TOT APC LOSS	144.03	285.13	522.34	13.73	2.13	11.08	36.08	72.58
29Y:	PCT APC LOSS	.07	.04	.15	.00	.00	.00	.00	.00
29Z:									
30A:	AVG HELO AUTH	797.00	.00	.00	29.50	240.00	.00	.00	.00
30B:	TOT HELO LOSS	11.34	1.12	3.20	1.37	1.95	.00	.00	.00
30C:	PCT HELO LOSS	.01	.00	.00	.05	.01	.00	.00	.00
30D:									
30E:	AVG ATM AUTH	2174.00	1204.00	7106.00	162.00	.00	.00	.00	.00
30F:	TOT ATM LOSS	.00	.00	45.87	.14	12.28	5.55	63.01	3.56
30G:	PCT ATM LOSS	.00	.00	.01	.00	.00	.00	.00	.00
30H:									
30I:									
30J:									
30K:									
30L:									
30M:									
30N:									
30O:									
30P:									
30Q:									
30R:									
30S:									
30T:									
30U:									
30V:									
30W:									
30X:									

372

1:	15	1	15	AVERAGE 15-DAY LOSS RATES **D-1 TO D-15**										
2:														
3:				.32	-.07	-.00	-.05	12.39	58.56	4.65	348.47	135.66	58.95	7.15
4:				.12	.11	.18	13.17	1.75	7.37	13.10	176.64	36.76	225.65	217.77
5:				.16	3.12	5.37	.75	.39						
6:				-.00	-.00	-.09	-.60	92.62	147.70	326.03	25.58	-.00	-.00	-.00
7:	09	CEM	01-	9	64.0	14.0	.0	10.0	24.0	36.0	72.0	72.0	18.0	
8:														
9:														
10:														
11:														
12:														
13:														
14:														
15:	15	16	30	AVERAGE 15-DAY LOSS RATES **D-16 TO D-30**										
16:				.28	-.07	-.61	-.10	5.42	9.98	34.50	102.00	70.93	26.13	47.82
17:				.17	-.12	-.26	7.53	2.06	8.29	89.24	58.41	14.75	149.91	110.82
18:				.30	12.65	8.70	.42	-.21						
19:				.00	-.00	-.09	-.06	38.72	54.12	105.59	11.38	-.00	-.00	-.00
20:	09	CEM	01-	9	56.0	14.0	122.0	70.0	34.0	52.0	60.0	42.0	18.0	
21:														
22:														
23:														
24:														
25:														
26:														
27:	30	1	30	AVERAGE 30-DAY LOSS RATES **D-1 TO D-90**										
28:	30	1	60											
29:	30	1	90											
30:														
31:				.40	-.09	-.50	-.12	-.00	-.00	-.00	-.00	-.00	-.00	-.00
32:				.23	-.15	-.33	-.10	-.00	-.00	-.00	-.00	-.00	-.00	-.00
33:				.27	-.00	-.00	.55	.27						
34:				-.00	-.00	-.14	-.07	-.00	-.00	-.00	-.00	-.00	-.00	-.00
35:	09	CEM	01-	9	40.0	9.0	50.0	12.0	23.0	33.0	27.0	27.0	14.0	
36:														
37:														
38:														
39:														
40:														
41:														
42:	30	91	12	AVE 30-DAY LOSS RATES **D-91 TO D-180**										
43:	30	91	15											
44:	30	91	19											
45:														
46:				.22	-.03	-.82	-.04	-.04	-.00	-.00	-.00	-.00	-.00	-.00
47:				.11	-.08	-.21	-.00	-.00	-.00	-.00	-.00	-.00	-.00	-.00
48:				.03	-.00	-.00	-.11	-.03						
49:														

373

50:	.60	.13	1.21	.15	.00	.00	.00	.00	.00	.00	.00	.00
51:	.29	.22	.45	.00	.00	.00	.00	.00	.00	.00	.00	.00
52:	.65	.00	.00	1.15	.59	.00	.00	.00	.00	.00	.00	.00
53:	.00	.00	.18	.13	.00	.00	.00	.00	.00	.00	.00	.00
60-ROSCM	01- 9	60.0	13.0	121.0	15.0	29.0	45.0	65.0	59.0	19.0		
63:												
64:												
65:												
66:												
67:												
68:												
69:												
70:	.50	.09	.72	.15	.00	.00	.00	.00	.00	.00	.00	.00
71:	.24	.18	.37	.00	.00	.00	.00	.00	.00	.00	.00	.00
72:	.28	.00	.00	.59	.23	.00	.00	.00	.00	.00	.00	.00
73:	.00	.00	.17	.03	.00	.00	.00	.00	.00	.00	.00	.00
74-ROSCM	01- 9	50.0	9.0	72.0	15.0	24.0	37.0	28.0	23.0	17.0		
75:												
76:												
77:												
78:												
79:												
80:												
81:												
82:	.27	.05	.00	.06	.00	.00	.00	.00	.00	.00	.00	.00
83:	.18	.10	.25	.00	.00	.00	.00	.00	.00	.00	.00	.00
84:	.00	.00	.00	.23	.06	.00	.00	.00	.00	.00	.00	.00
85:	.00	.00	.10	.02	.00	.00	.00	.00	.00	.00	.00	.00
86-ROSCM	01- 9	27.0	5.0	.0	6.0	14.0	25.0	10.0	6.0	10.0		
87:												
88:												
89:												
90:												
91:												
92:												
93:												
94:	.19	.04	.00	.05	.00	.00	.00	.00	.00	.00	.00	.00
95:	.13	.07	.16	.00	.00	.00	.00	.00	.00	.00	.00	.00
96:	.04	.00	.00	.16	.05	.00	.00	.00	.00	.00	.00	.00
97:	.00	.00	.06	.01	.00	.00	.00	.00	.00	.00	.00	.00
98-ROSCM	01- 9	13.0	4.0	.0	5.0	13.0	16.0	4.0	5.0	6.0		
99:												
100:												
101:												
102:												
103:												
104:												
105:	.30	.03	.74	.04	.00	.00	.00	.00	.00	.00	.00	.00
106:	.14	.10	.33	.00	.00	.00	.00	.00	.00	.00	.00	.00
107:	.03	.00	.00	.12	.01	.00	.00	.00	.00	.00	.00	.00
108:	.00	.00	.04	.00	.00	.00	.00	.00	.00	.00	.00	.00
109:												
110-ROSCM	01- 9	30.0	3.0	74.0	4.0	14.0	33.0	3.0				

374

## CHAPTER 17

### Utility - COUNT/DIVISIONS

**17.1 DESCRIPTION OF PROCESSING:** This program performs computations on the input data in addition to logic tested read and write statements.

**17.1.1 PURPOSE/FUNCTIONS:** The purpose of this utility is to produce the COUNT/DIVISIONS file. The COUNT/DIVISIONS file summarizes:

- o The number of divisions arriving in the theater period by period for the seven time periods of the analysis.
- o The average number of divisions arriving period by period.
- o The total number of U.S. divisions in theater on D-Day.
- o The average number of divisions in theater period by period.

The utility uses as its source of information the 73CEM5.BLUE/P88-XX file which is produced by CEM. This file provides information on each division participating in the theater conflict, such as its status and arrival date in theater.

In that the force structures, status and arrivals of divisions in the theater are relatively constant from each run of the CEM. This utility need only be run once during a study. The COUNT/DIVISIONS file will be used as one of the four input files into the following utility, CONTROL/COMPILER.

**17.1.2 PROGRAM INPUT/OUTPUT STRUCTURE:** The overall structure of the utility is pictured in FIGURE III.17.1. The logic flow of the utility is depicted in FIGURE III.17.2. The source code of the utility is listed in FIGURE III.17.3.

**17.1.2.A INPUT DATA AND DATA BASE:** There is one input file to this utility. The file is a product of the CEM model and describes the relative sizes and arrivals of specific divisions in the theater. In the current example the file is cataloged as an element under the CEM program file 73CEM5 using the element name BLUE/P88-XX. The proper program file and element names can be obtained from the CEM Operator/Analyst. Volume I of this documentation details a record layout for the file.

**17.1.2.B OUTPUT DATA AND DATA FILES:** This utility produces one output file referred to as COUNT/DIVISIONS-XX. The file will be cataloged as an element under the current study's program file (e.g., SECRET\*82WARFP88). The file will contain a summary of:

- o The total number of divisions arriving in the theater time period by time period.
- o The average number of divisions arriving in the theater by period.

- o The total number of divisions in the theater on D-Day.
- o The average number of divisions in the theater by period.

Figure III.17.4 presents an example of the data contained in the output file.

17.1.2.C DATA VARIABLE DICTIONARY: The following section identifies and defines all variables used in this utility.

<u>NAME</u>	<u>DEFINITION</u>
TITLE	An alphanumeric variable used to identify the type unit record that has been read from the 73CEM5.BLUE/P88-XX input file.
DIV1, DIV2	These two alphanumeric variables are single dimension arrays. DIV1 will contain the first 6 characters and DIV2 the last 3 characters of the Division name. These values will be read from the input file.
CODE1	An alphanumeric variable which indicates whether the Brigade is active or a ghost.  A = Active Not-A = Ghost (Reserve)
SEC1	This integer variable is a single dimension array in the utility. It denotes the first (i.e., X) coordinate of the FEBA sector. Valid entries are 000-999.
SEC2	This array denotes the second (i.e., Y) coordinate of the FEBA sector. Valid entries are 000-999. This value is read from the input file.
CODE	This is a single dimension array which indicates whether or not the unit is active or reserve. An A = Active, Not A = Inactive.
TYPE	One character integer variable used as a single dimension array. This variable is used to indicate whether the Blue unit is a U.S. unit or an allied unit.  If TYPE = 1, the unit is U.S. If TYPE NOT = 1, the unit is an allied unit.  This data is read from the input file.
NUM	A 3 character alphanumeric variable. This variable is used by the utility to identify and screen units that will not be counted. If the unit's number is equal to "100" the unit will continue to be analyzed. If it is not

equal to "100" it is eliminated.

BDES	An integer variable used as a single dimension array in the utility. This variable is used to indicate the number of Brigades in a specific Division. The subscript for BDES must be consistent with the current Division. This value is calculated in the utility.
CYCLE	A 2 character integer value which is used by the utility and the CEM as a unit of measure for time. A CYCLE = 4 days. It is read in from the input file. It is used to identify day and time period specific divisions arrive in the heater. As is currently structured, valid entries for CYCLE are from 1 to 45.
ARR1, ARR2	These two, 2 character integer variables identify the first (ARR1) and second (ARR2) days of the CYCLE of the arriving units. These variables are read from the input file. These values will be assigned to positions in the 2 dimensional matrix ARRIVE discussed below.
ARRIVE	This is a 2 dimensional matrix populated by 2 character integer values. There will be 7 occurrences of the "X" axis of the matrix (1 occurrence for each of the 7 time periods) and 30 occurrences of the "Y" axis of the matrix (1 occurrence for each of the maximum 30 days in a period.) It should be noted that in the first 2 periods of the study will have only 15 days. Entries in this matrix will be assigned the values found in the ARR1 and ARR2 variables from the current input record. The assignment of the values to the proper entry in the matrix will be determined by the utility and depend upon the value of the variable CYCLE of the arriving Division.
IP, IPP	Integer variables used by the utility to identify 1 of the 7 time periods in the ARRIVE matrix.
CHECK	This is an integer variable used in the utility as a single dimension array with 7 occurrences. Each entry in the array is calculated by the utility and reflects the total number of divisions arriving in the theater during each of the 7 time periods of the study.
ICOUNT	This is an integer variable used to temporarily hold the number of divisions arriving during a period i.e., ARRIVE (X,Y). It is used as the upper limit for a DO LOOP.
PERIOD	This is a real variable which is dimensioned as an array with 7 occurrences, 1 occurrence for each of the 7 time periods in the study. Each entry in the array

reflects for each time period the average number of divisions arriving in theater.

I	A counter used to accumulate the number of Division records read from the input file which are to be included in the calculating of Division strengths.
DDAY	An integer variable used to count the number of divisions in theater on D-DAY.
JI, JK, JL, LJ, LK, JC, JB, KJ, KM, IB, ICD, JJJ	Various integer variables used as counters and subscripts in the utility.

17.2     OPERATING ENVIRONMENT: This program is implemented on the EXECUTIVE-8 operating system.

17.2.1   SUPPORT SOFTWARE: The utility requires the FORTRAN IV compiler and the UNIVAC 1100/82 system software for its execution.

17.2.2   I/O DEVICE REQUIREMENTS: This utility uses one file as input which is maintained on disk. It will in turn produce an output file which will also be maintained on disk.

17.3     MAINTENANCE PROCEDURES: The program is maintained by the MPP analyst.

17.3.1   PROGRAMMING CONVENTIONS: The utility is written in FORTRAN IV and follows FORTRAN conventions. The utility is structured to translate the arrival dates of divisions in-theater from theater cycles (i.e. 4 day periods) to the appropriate day within the appropriate time period used in the study.

17.3.2   INTERNAL ERROR ROUTINES: There are no explicit error handling routines written into the utility. As a result the only errors detected by the system will be identified. These errors must be corrected using system documentation.

# COUNT/DIVISIONS STRUCTURE

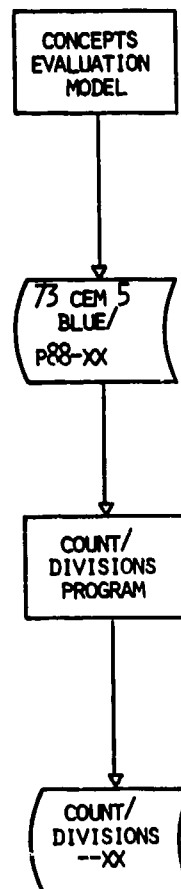


Figure III.17.1

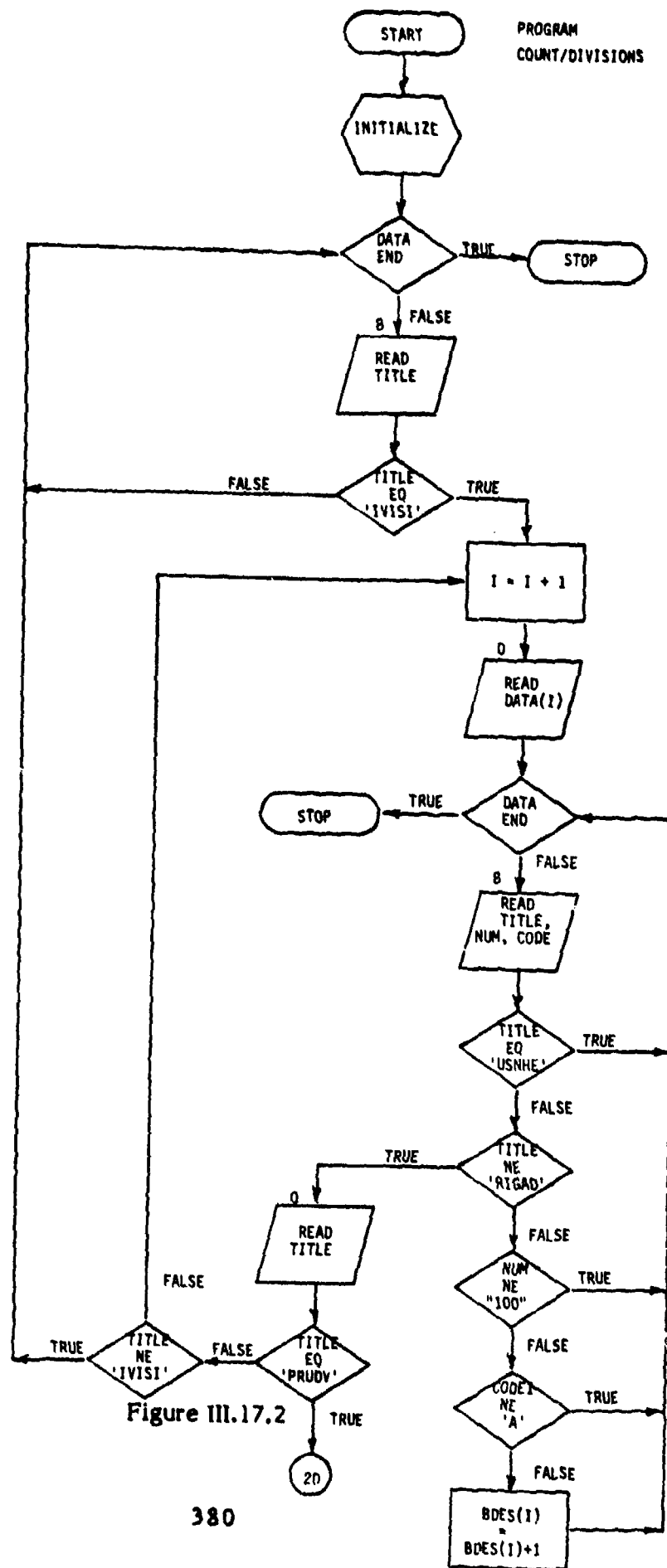


Figure III.17.2

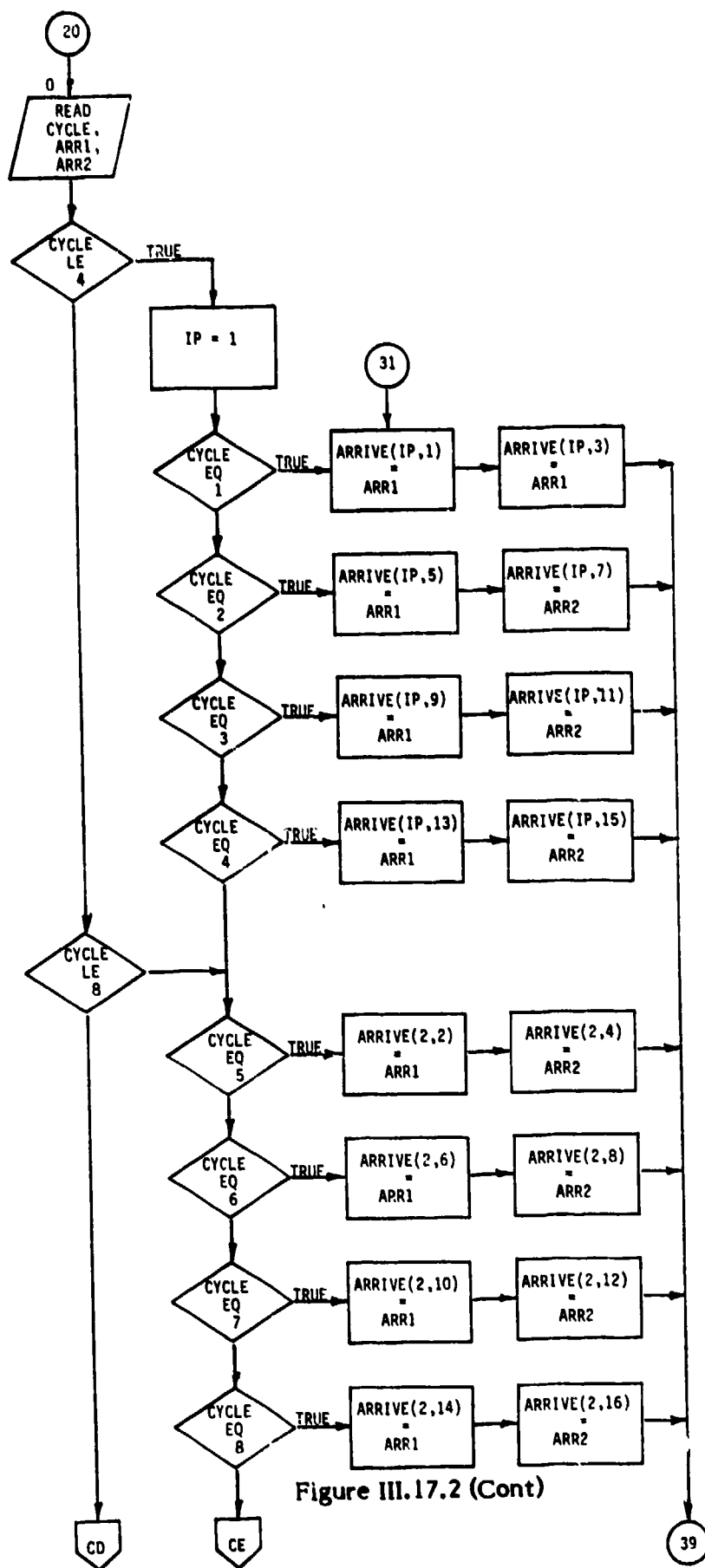


Figure III.17.2 (Cont)

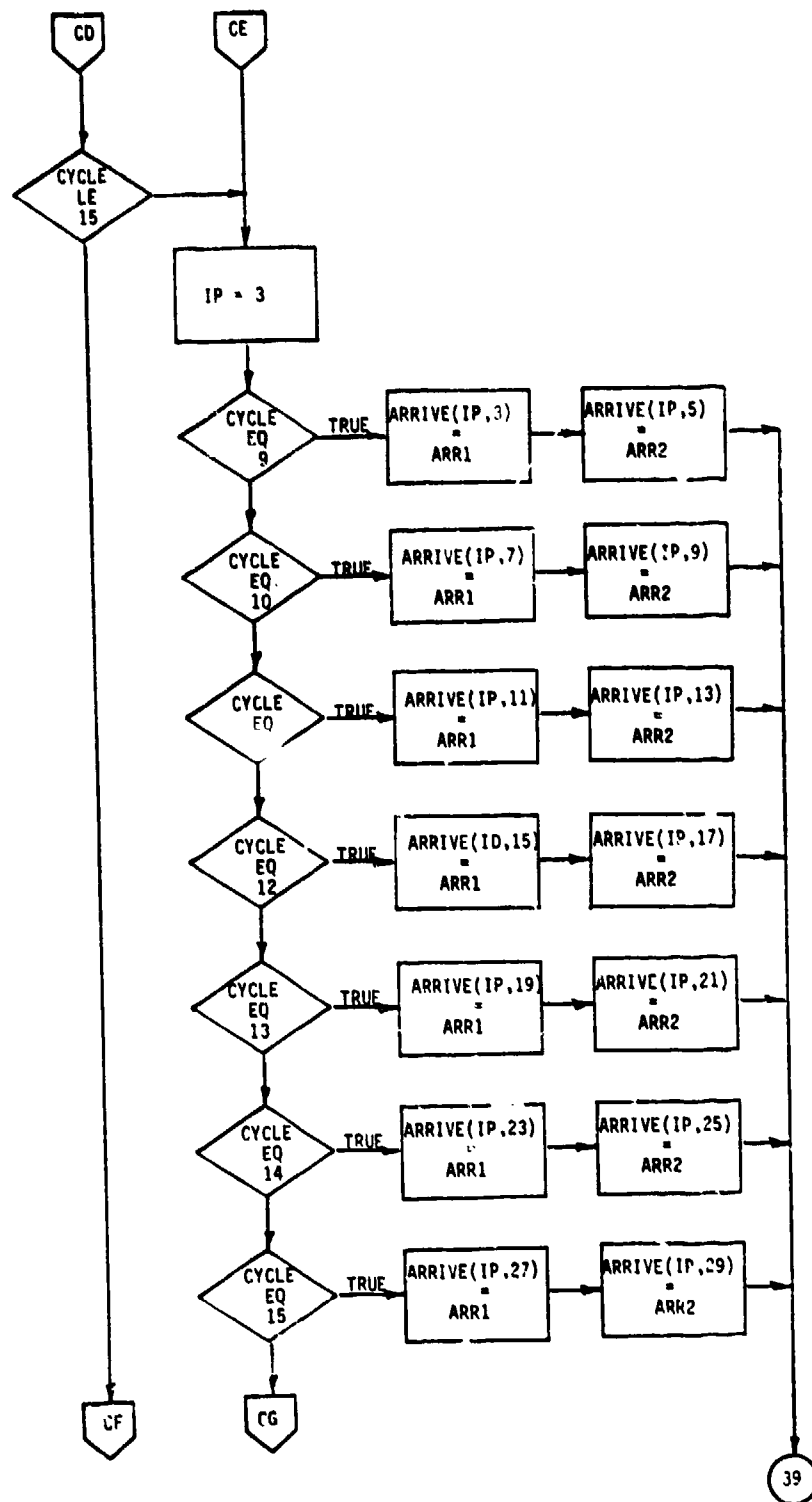


Figure III.17.2 (Cont)

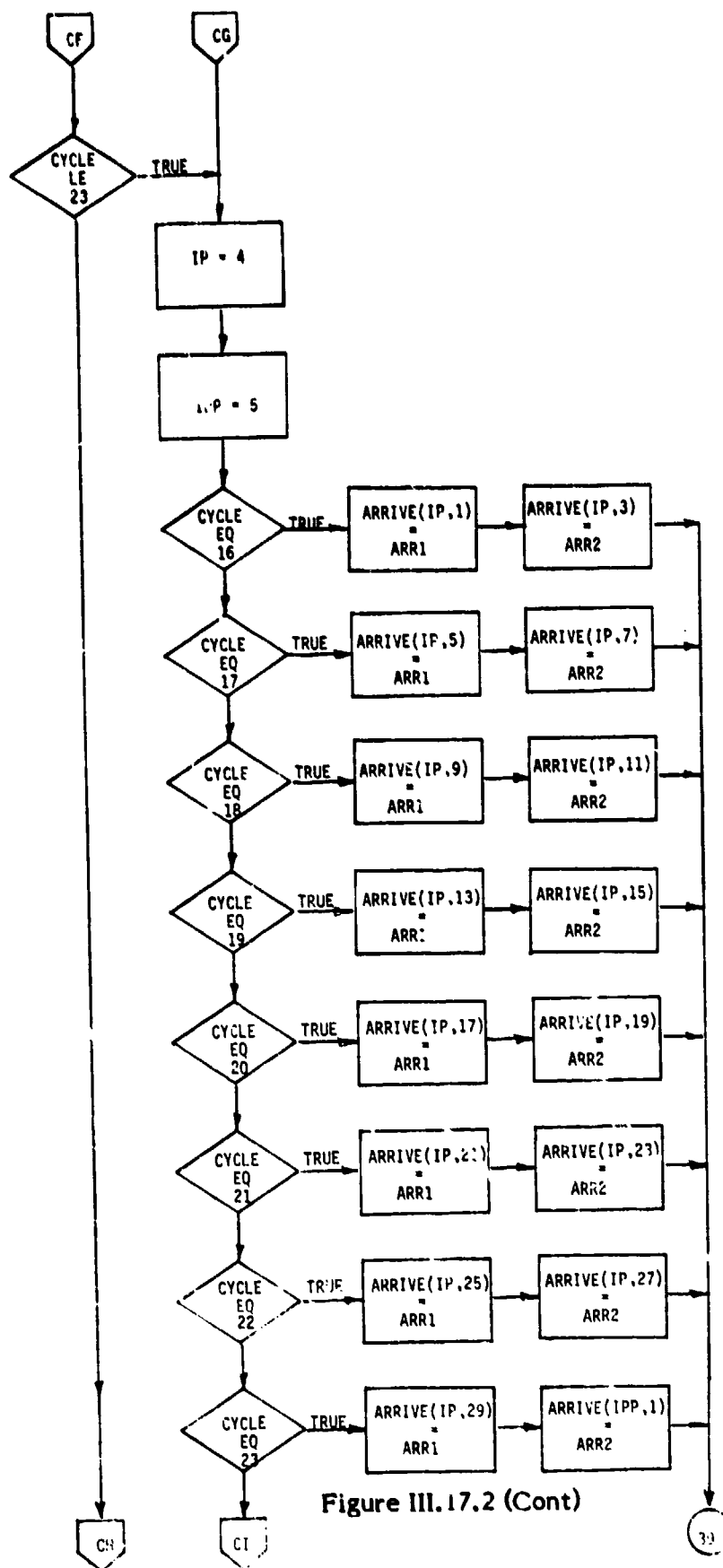


Figure III.17.2 (Cont)

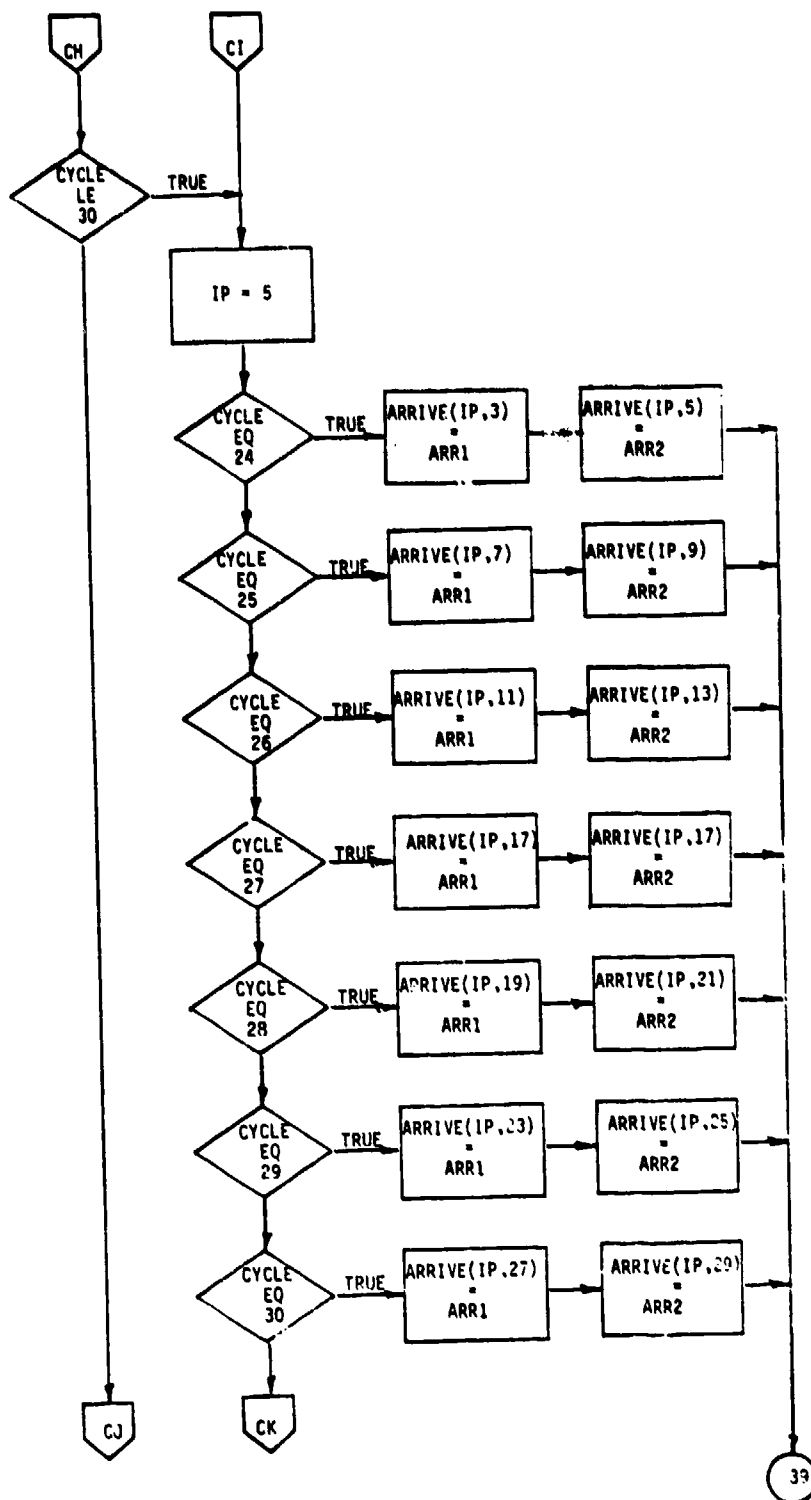


Figure III.17.2 (Cont)

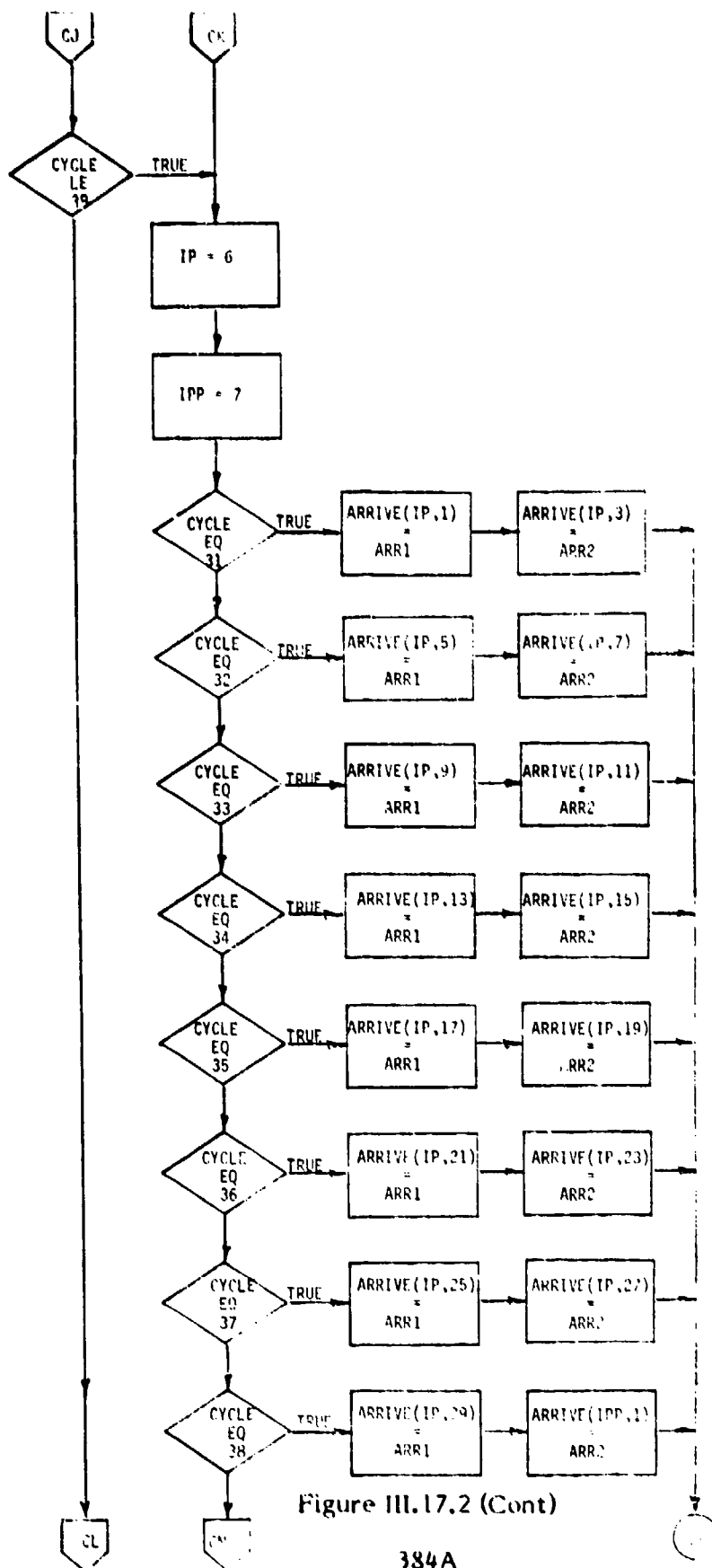


Figure III.17.2 (Cont)

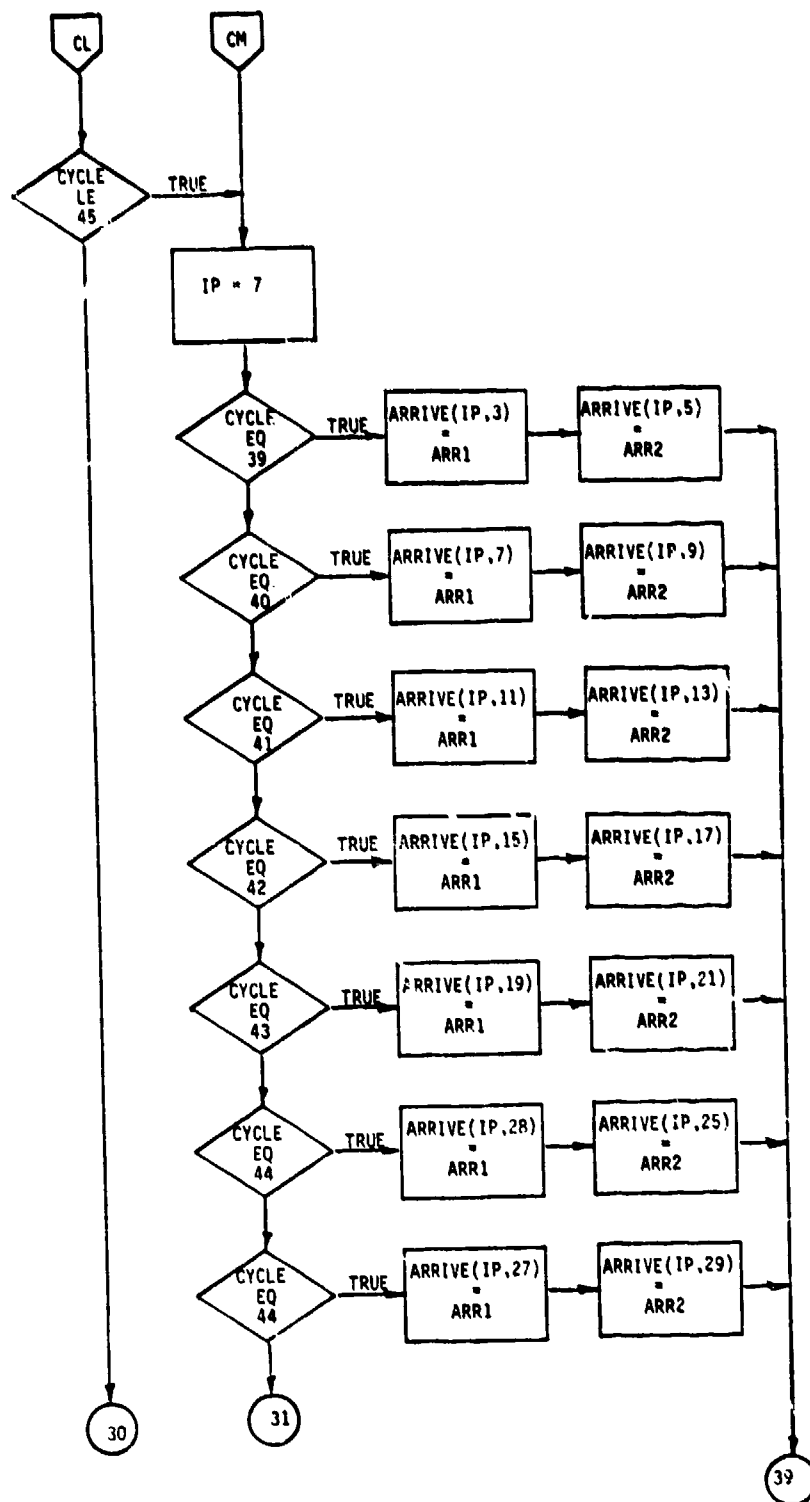


Figure III.17.2 (Cont)

384B

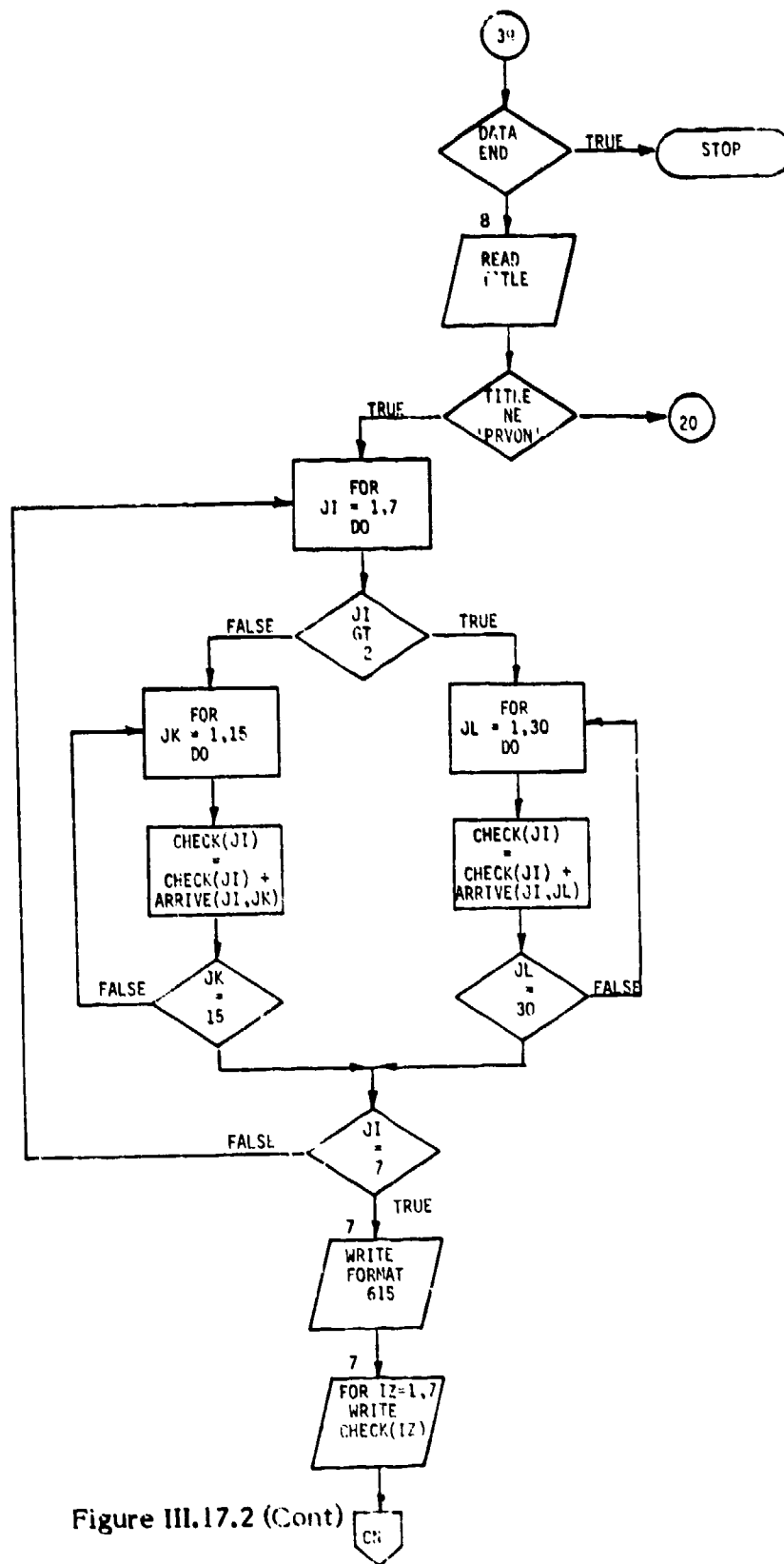
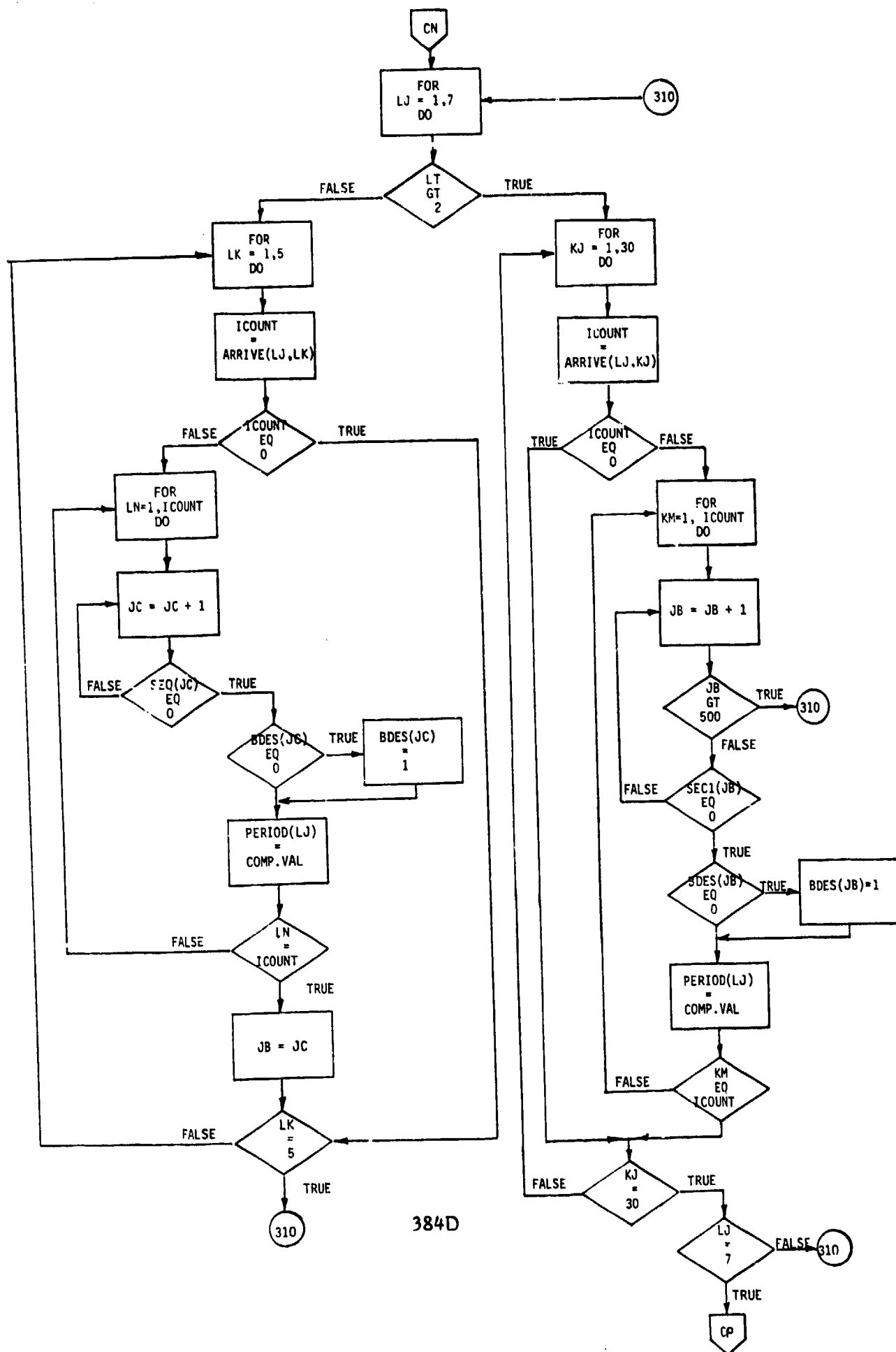


Figure III.17.2 (Cont)



384D

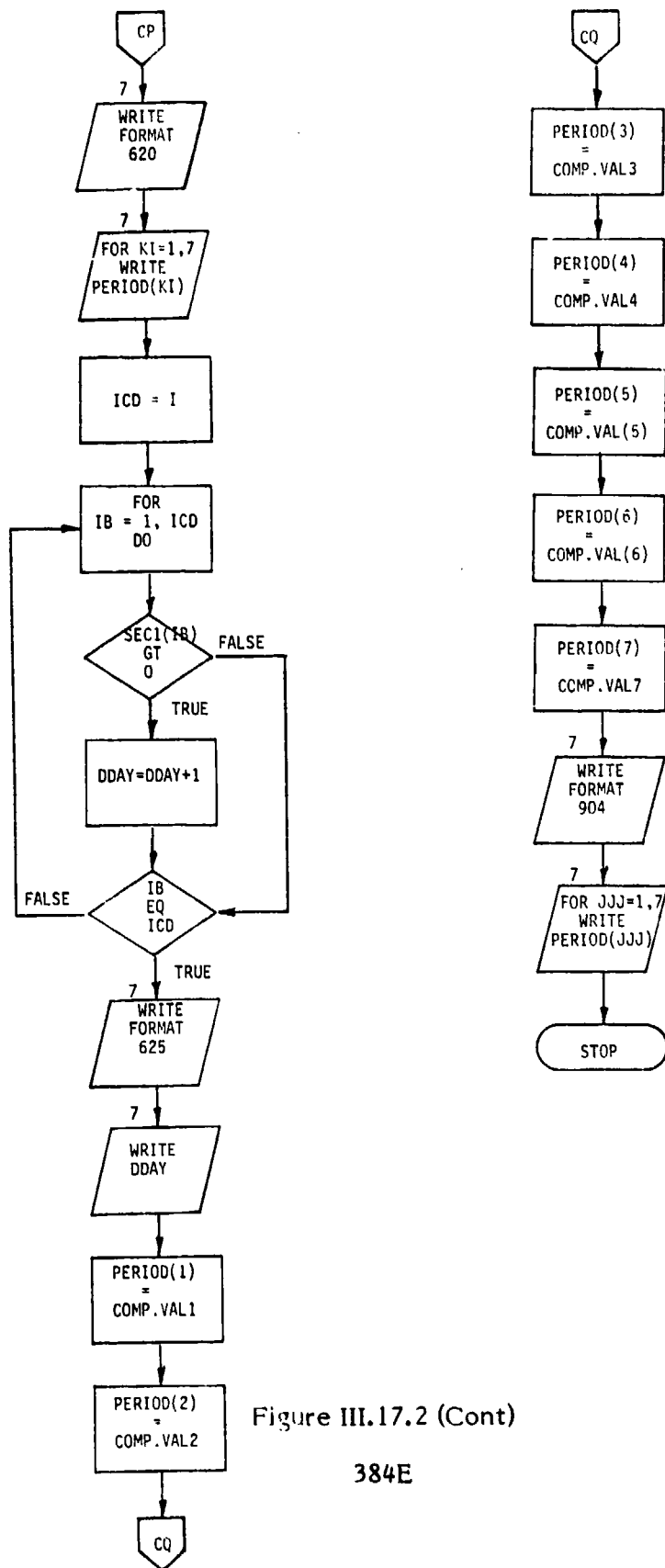


Figure III.17.2 (Cont)

384E

**THIS PAGE IS INTENTIONALLY LEFT BLANK**

**384F**

UNCLASSIFIED\*\*\*FILE NAME:12701 ELEMENT NAME:COUNT/DIVISIONS\*\*\*UNCLASSIFIED

```

1:C*****
2:C*****
3:C*****THIS PROGRAM IS WRITTEN IN FORTRAN IV.
4:C*****
5:C*****
6:C*****PURPOSE OF PROGRAM:
7:C*****
8:C*****THIS UTILITY IS DESIGNED TO READ THE BLUE FORCE INPUT DATA FILE
9:C*****TO CEM AND PROVIDE THE AVERAGE US FORCE DIVISIONAL COUNT FOR
10:C*****FOR EACH OF THE SEVEN TIME PERIODS. THE OUTPUT OF THIS UTILITY
11:C*****WILL BE USED AS INPUT TO THE UTILITY "62XQT.CONTROL/COMPILER.
12:C*****
13:C*****
14:C*****VARIABLE DICTIONARY
15:C*****
16:C*****VARIABLE NAME      DEFINITION
17:C*****TITLE              UNIT IDENTIFIER(I.E. DIVISION,BRIGADE)
18:C*****DIV1(I)             6 CHARACTERS OF THE DIVISION NAME
19:C*****DIV2(I)             3 CHARACTERS OF THE DIVISION NAME
20:C*****CODE(I)            IDENTIFIES UNIT STATUS (I.E.ACTIVE
21:C                          RESERVE)
22:C*****SEC1(I)             FIRST COORDINATES OF FEBA SECTOR
23:C*****SEC2(I)            SECOND SET OF COORDINATES OF THE
24:C                          FEBA SECTOR
25:C*****TYPE(I)            CEM PARTITION BY US AND OTHER COUNTRIES
26:C*****ARRIVE(I,J)        ARRAY FOR STORING WHEN UNITS ARRIVE IN
27:C                          THEATER BY TIME PERIOD/DAY OF PERIOD
28:C*****CYCLE               CYCLE OF THE CEM WHICH IS EQUAL TO 4 DAYS
29:C*****ARR1                THE FIRST DAY OF THE CYCLE OF ARRIVING UNITS
30:C*****ARR2                THE 2ND DAY OF THE CYCLE OF ARRIVING UNITS
31:C*****ICD                TOTAL NUMBER OF BLUE DIVISIONS IN FORCE
32:C*****IP                 PERIOD COUNTER USED IN ARRAYS
33:C*****IDP                PERIOD COUNTER USED IN ARRAYS
34:C*****BDES(I)            NUMBER OF BRIGADES IN DIVISION D
35:C*****SIZE                DIVISION'S TRUE SIZE AS A PERCENT. MUST
36:C                          HAVE 3 BRIGADES TO = 1
37:C*****ODAY               NUMBER OF DIVISIONS IN THEATER ON D-DAY
38:C*****CHECK(I)           THE NUMBER OF DIVISIONS ARRIVING DURING
39:C                          PERIOD I
40:C*****COUNT(I)          COUNTER USED TO INSURE CHECK IS NOT
41:C                          EXCEEDED
42:C*****CODE1              USED TO CHECK ON WHETHER A BRIGADE IS AN
43:C                          ACTIVE OR GHOST ORGANIZATION
44:C*****PERIOD(I)           AVERAGE DIVISION STRENGTH FOR PERIOD P
45:C                          FINAL PRODUCE
46:C*****
47:C*****READING INTO ARRAYS ALL DIVISIONS AND THEIR NUMBER OF BRIGADES
48:C*****IN THE BLUE FORCE.
49:C*****
50:      DIMENSION DIV1(500),DIV2(500),CODE(500),PERIOD(7)
51:      REAL AUN
52:      INTEGER SEC1(500),SEC2(500),TYPE(500),ARRIVE(7,30),ARR1,ARR2,
53:      ICHECK(7),ODAY,BDES(500),CYCLE
54:10  READ(8,100,FND=99)TITLE
55:100  FORMAT(1X,A')
56:      IF(TITLE.EQ.'DIVISI')GO TO 11
57:      GO TO 10

```

Figure III.17.3

UNCLASSIFIED\*\*\*FILL NAME:R2X0T ELEMENT NAME:COUNT/DIVISIONS\*\*\*UNCLASSIFIED

```

58:17 READ(0,100)TITLE
59: IF(TITLE.EQ.'RRVDR')GO TO 20
60: IF(TITLE.NE.'IVISI')GO TO 10
61:11 I=1
62: READ(0,101)DIV1(1),DIV2(1),SEC1(1),SEC2(1),CODE(1),TYPE(1)
63:101 FORMAT(10X,F6,A3,3X,13,2X,13,1X,A1,21X,11)
64:13 READ(0,102,FND=99)TITLE,NUM,CODE1
65:102 FORMAT(1X,A1,A6,A7,4X,A1)
66: IF(TITLE.EQ.'VSNHE')GO TO 13
67: IF(TITLE.NE.'RIGAD')GO TO 12
68: IF(NUM.NE.'100')GO TO 13
69: IF(CODE1.NE.'A')GO TO 13
70: DOES(1)=DOES(1)+1
71: GO TO 13
72:C*****
73:C*****HEADING INTO ARRAYS BY TIME PERIOD AND DAY THE NUMBER OF
74:C*****ARRIVING DIVISIONS.
75:C*****
76:20 READ(0,200)CYCLE,APR1,APR2
77:200 FORMAT(10X,I4,4X,I2,2X,I2)
78: IF(CYCLE.LE.4)GO TO 30
79: IF(CYCLE.LE.6)GO TO 40
80: IF(CYCLE.LE.8)GO TO 50
81: IF(CYCLE.LE.13)GO TO 60
82: IF(CYCLE.LE.30)GO TO 70
83: IF(CYCLE.LE.35)GO TO 80
84: IF(CYCLE.LE.45)GO TO 90
85:20 I=1
86: IF(CYCLE.EQ.1)GO TO 31
87: IF(CYCLE.EQ.2)GO TO 32
88: IF(CYCLE.EQ.3)GO TO 33
89: IF(CYCLE.EQ.4)GO TO 34
90:40 IF(CYCLE.EQ.5)GO TO 41
91: IF(CYCLE.EQ.6)GO TO 42
92: IF(CYCLE.EQ.7)GO TO 43
93: IF(CYCLE.EQ.8)GO TO 44
94:50 I=3
95: IF(CYCLE.EQ.9)GO TO 51
96: IF(CYCLE.EQ.10)GO TO 52
97: IF(CYCLE.EQ.11)GO TO 53
98: IF(CYCLE.EQ.12)GO TO 54
99: IF(CYCLE.EQ.13)GO TO 55
100: IF(CYCLE.EQ.14)GO TO 56
101: IF(CYCLE.EQ.15)GO TO 57
102:60 I=4
103: IF(I=5)
104: IF(CYCLE.EQ.16)GO TO 31
105: IF(CYCLE.EQ.17)GO TO 32
106: IF(CYCLE.EQ.18)GO TO 33
107: IF(CYCLE.EQ.19)GO TO 34
108: IF(CYCLE.EQ.20)GO TO 35
109: IF(CYCLE.EQ.21)GO TO 36
110: IF(CYCLE.EQ.22)GO TO 37
111: IF(CYCLE.EQ.23)GO TO 36
112:70 I=5
113: IF(CYCLE.EQ.24)GO TO 51
114: IF(CYCLE.EQ.25)GO TO 52

```

Figure III.17.3 (Cont)

UNCLASSIFIED\*\*\*FILE NAME:R2X01 ELEMENT NAME:COUNT/DIVISIONS\*\*\*UNCLASSIFIED

```

115:      IF(CYCLE.EC.25)GO TO 53
116:      IF(CYCLE.EC.27)GO TO 54
117:      IF(CYCLE.EC.28)GO TO 55
118:      IF(CYCLE.EC.29)GO TO 56
119:      IF(CYCLE.EC.37)GO TO 57
120:50      IP=6
121:      IPP=7
122:      IF(CYCLE.EC.31)GO TO 31
123:      IF(CYCLE.EC.32)GO TO 32
124:      IF(CYCLE.EC.33)GO TO 33
125:      IF(CYCLE.EC.34)GO TO 34
126:      IF(CYCLE.EC.35)GO TO 35
127:      IF(CYCLE.EC.36)GO TO 36
128:      IF(CYCLE.EC.37)GO TO 37
129:      IF(CYCLE.EC.38)GO TO 38
130:90      IP=7
131:      IF(CYCLE.EC.39)GO TO 51
132:      IF(CYCLE.EC.40)GO TO 52
133:      IF(CYCLE.EC.41)GO TO 53
134:      IF(CYCLE.EC.42)GO TO 54
135:      IF(CYCLE.EC.43)GO TO 55
136:      IF(CYCLE.EC.44)GO TO 56
137:      IF(CYCLE.EC.44)GO TO 57
138:31      ARRIVE(IP,1)=ARR1
139:      ARRIVE(IP,2)=ARR2
140:      GO TO 39
141:32      ARRIVE(IP,5)=ARR1
142:      ARRIVE(IP,7)=ARR2
143:      GO TO 39
144:33      ARRIVE(IP,4)=ARR1
145:      ARRIVE(IP,11)=ARR2
146:      GO TO 39
147:34      ARRIVE(IP,17)=ARR1
148:      ARRIVE(IP,15)=ARR2
149:      GO TO 39
150:35      ARRIVE(IP,17)=ARR1
151:      ARRIVE(IP,15)=ARR2
152:      GO TO 39
153:36      ARRIVE(IP,21)=ARR1
154:      ARRIVE(IP,22)=ARR2
155:      GO TO 39
156:37      ARRIVE(IP,25)=ARR1
157:      ARRIVE(IP,27)=ARR2
158:      GO TO 39
159:38      ARRIVE(IP,25)=ARR1
160:      ARRIVE(IPP,1)=ARR2
161:      GO TO 39
162:41      ARRIVE(I2,2)=ARR1
163:      ARRIVE(I2,4)=ARR2
164:      GO TO 39
165:42      ARRIVE(I2,4)=ARR1
166:      ARRIVE(I2,4)=ARR2
167:      GO TO 39
168:43      ARRIVE(I2,10)=ARR1
169:      ARRIVE(I2,10)=ARR2
170:      GO TO 39
171:44      ARRIVE(I2,14)=ARR1

```

Figure III.17.3 (Cont)

UNCLASSIFIED\*\*FILE NAME:62X01 ELEMENT NAME:COUNT/DIVISIONS\*\*UNCLASSIFIED

```

172:      ARRIVE(3,1)=ARR2
173:      GO TO 39
174:51     ARRIVE(1P,3)=ARR1
175:      ARRIVE(1P,5)=ARR2
176:      GO TO 39
177:52     ARRIVE(1P,7)=ARR1
178:      ARRIVE(1P,9)=ARR2
179:      GO TO 39
180:53     ARRIVE(1P,11)=ARR1
181:      ARRIVE(1P,13)=ARR2
182:      GO TO 39
183:54     ARRIVE(1P,15)=ARR1
184:      ARRIVE(1P,17)=ARR2
185:      GO TO 39
186:55     ARRIVE(1P,19)=ARR1
187:      ARRIVE(1P,21)=ARR2
188:      GO TO 39
189:56     ARRIVE(1P,23)=ARR1
190:      ARRIVE(1P,25)=ARR2
191:      GO TO 39
192:57     ARRIVE(1P,27)=ARR1
193:      ARRIVE(1P,29)=ARR2
194:39     READ(8,100,END=99)TITLE
195:      IF(TITLE.NE.'RRVDV')GO TO 300
196:      GO TO 20
197:300     CONTINUE
198:C*****
199:C*****TOTALING UP BY TIME PERIOD THE NUMBER O ARRIVING DIVISIONS
200:C*****
201:      GO 301 JI=1,7
202:      IF(JI.GT.2)GO TO 303
203:      GO 304 JK=1,15
204:      CHECK(JI)=CHECK(JI)+ARRIVE(JI,JK)
205:304     CONTINUE
206:      GO TO 301
207:303     GO 305 JL=1,30
208:      CHECK(JI)=CHECK(JI)+ARRIVE(JI,JL)
209:305     CONTINUE
210:301     CONTINUE
211:      WRITE(7,615)
212:615     FORMAT(' TOTAL NUMBER OF ARRIVING DIVISIONS BY PERIOD')
213:      WRITE(7,612)(CHECK(12),12=1,7)
214:612     FORMAT(1X,7(13,1X))
215:C*****
216:C*****COMPUTING PERCENT OF TIME EACH ARRIVING DIVISION SPENT IN THE
217:C*****PERIOD DURING WHICH IT ARRIVED
218:C*****
219:      GO 310 LJ=1,7
220:      IF(LJ.GT.2)GO TO 311
221:      GO 312 LK=1,15
222:      ICOUNT=ARRIVE(LJ,LK)
223:      IF(ICOUNT.EQ.0)GO TO 317
224:      GO 313 LV=1,ICOUNT
225:314     JC=JC+1
226:      IF(SEC(1(JC).EQ.0.AND.CODE(JC).EQ.'A'.AND.TYPE(JC).EQ.1)
227:      GO TO 320
228:      GO TO 314

```

Figure III.17.3 (Cont)

UNCLASSIFIED\*\*\*FILE NAME:2XCT ELEMENT NAME:COUNT/DIVISIONS\*\*UNCLASSIFIED

```

229:320 IF(RDES(JC).EQ.D)RDES(JC)=1
230: PERIOD(LJ)=PERIOD(LJ)+(RDES(JC)/3.D)*((15-LK+1)/15.D)
231:313 CONTINUE
232: JH=JC
233:312 CONTINUE
234: GO TO 310
235:311 GO 315 KJ=1,30
236: ICOUNT=ARRIVE(LJ,KJ)
237: IF(ICOUNT.EQ.D)GO TO 315
238: GO 316 KM=1,ICOUNT
239:317 JH=JH+1
240: IF(JH.LT.50)GO TO 310
241: IF(SEC(JR).EQ.D.AND.CODE(JR).EQ.'A'.AND.TYPE(JB).EQ.1)
242: GO TO 321
243: GO TO 317
244:321 IF(RDES(JR).EQ.D)RDES(JR)=1
245: PERIOD(LJ)=PERIOD(LJ)+(RDES(JR)/3.D)*((15-KJ+1)/30.D)
246:316 CONTINUE
247:315 CONTINUE
248:310 CONTINUE
249: WRITE(7,620)
250:620 FORMAT(' AVERAGE NUMBER OF ARRIVING DIVISIONS BY PERIOD')
251: WRITE(7,619)(PERIOD(KI),KI=1,7)
252:619 FORMAT(7(1X,F6.3))
253:C*****
254:C*****COMPUTING THE NUMBER OF DIVISIONS IN THEATER ON U-DAY
255:C*****
256: ICD=1
257: GO 500 ID=1,ICD
258: IF(SEC(IF).LT.D.AND.CODE(IF).EQ.'A'.AND.TYPE(IE).EQ.1)
259: GO TO 771
260: GO TO 500
261:771 UDAY=UDAY+1
262:500 CONTINUE
263: WRITE(7,625)
264:625 FORMAT(' NUMBER OF US DIVISIONS IN THEATER ON U-DAY')
265: WRITE(7,627)UDAY
266:627 FORMAT(1X,17)
267:C*****
268:C*****COMPUTING THE FINAL AVERAGE DIVISIONAL COUNT FOR EACH PERIOD
269:C*****
270: PERIOD(1)=PERIOD(1)+UDAY
271: PERIOD(2)=PERIOD(2)+CHECK(1)+UDAY
272: PERIOD(3)=PERIOD(3)+CHECK(1)+CHECK(2)+UDAY
273: PERIOD(4)=PERIOD(4)+CHECK(1)+CHECK(2)+CHECK(3)+UDAY
274: PERIOD(5)=PERIOD(5)+CHECK(1)+CHECK(2)+CHECK(3)+CHECK(4)+UDAY
275: PERIOD(6)=PERIOD(6)+CHECK(1)+CHECK(2)+CHECK(3)+CHECK(4)+CHECK(5)
276: 1+UDAY
277: PERIOD(7)=PERIOD(7)+CHECK(1)+CHECK(2)+CHECK(3)+CHECK(4)+CHECK(5)
278: 1+CHECK(6)+UDAY
279: WRITE(7,904)
280:904 FORMAT(' THE AVERAGE DIVISIONAL COUNT BY PERIOD')
281: WRITE(7,905)(PERIOD(JJJ),JJJ=1,7)
282:905 FORMAT(1X,7(F4.1,1X))
283:99 STOP
284: END

```

Figure III.17.3 (Cont)

UNCLASSIFIED\*\*\*EXAMPLE OF THE OUTPUT FROM UTILITY COUNT/DIVISIONS\*\*\*UNCLASSIFIED

1: TOTAL NUMBER OF ARRIVING DIVISIONS BY PERIOD.  
2: 8 2 0 0 0 0 0  
3: AVERAGE NUMBER OF ARRIVING DIVISIONS BY PERIOD.  
4: 6.062 1.500 0.000 0.000 0.000 0.000 0.000  
5: NUMBER OF US DIVISIONS IN THEATER ON D-DAY.  
6: 42  
7: THE AVERAGE DIVISIONAL COUNT BY PERIOD.  
8: 48.1 51.5 52.0 52.0 52.0 52.0 52.0

Figure III.17.4

## CHAPTER 18

### Utility - CONTROL/COMPILER

**18.1 DESCRIPTION OF PROCESSING:** This program performs computations in addition to the logic tested read and write statements.

**18.1.1 PURPOSE/FUNCTIONS:** The purpose of this utility is to gather and organize data from five input files and produce the CONTROL/XX file. This file will be used in conjunction with the current ITMID/FINAL file, which was produced by the earlier ITMID/REC-A utility, as of two input files to the Equipment Loss Consolidator (ELCON) program which will be subsequently executed.

**18.1.2 PROGRAM INPUT/OUTPUT STRUCTURE:** The overall structure of the utility is pictured in FIGURE III.18.1. The logic flow followed by the utility is depicted in FIGURE III.18.2. The source code of the utility is listed in FIGURE III.18.3.

**18.1.2.A INPUT DATA AND DATA BASE:** The source of data for this utility will be provided by a collection of five existing files. Three of the files will have been automatically produced by other MPP utilities, which must have been successfully completed prior to the execution of this utility. These three files are COUNT/DIVISIONS-XX, SCENARIO/XX, and CEM/LOSSES. The remaining two files, CONTROL/TEMP, and REDARTY/DEGR-XX are manually created by the user via the system editor. Source document for the CONTROL/TEMP file will be the previous study's control file; the source document for the REDARTY/DEGR-XX file will be the LOG REPORT produced by the CEM. Each input file is discussed below.

- o CONTROL/TEMP - As noted above this file is created manually by the user, using the previous study's CONTROL file as its' source. This file will provide to the output file both its form and selected elements of information such as:
  - oo run parameters or control data,
  - oo Number of days in each period,
  - oo For each of the loss rates specified, the first and the last time period considered.
  - oo Inter-theater and intra-theater transportation loss fraction by air, sea and LOC.
  - oo For each of the 4 combat postures, the fractional daily loss for each of the 22 vulnerability categories.

The remaining records inserted into this file will be zeroed out. These records will be completed by the utility using data extracted from the remaining four input files. Figure III.18.4 depicts this input data file.

- o COUNT/DIVISIONS - This is the second of the five input files to this utility. It will supply to the utility the number of Blue divisions in theater on a period by period basis. Figure III.18.5 presents this data file.

- o SCENARIO/XX - This is the third file to be used by the utility. This file was produced by the SEARCH/ENGAGEREP utility and denotes for each of the four postures, the daily, fractional loss suffered in each of the 22 vulnerability categories. Figure III.18.6 presents an example of this input data file.
- o REDARTY/DEGR-XX - This is the fourth file to be used by this utility. This file was created manually by the study analyst using data supplied by the CEM LOG REPORT on RED LOGISTICS EFFECTIVENESS. From this data the utility will record the fraction of full strength Red artillery effectiveness to be applied by time period. Figure III.18.7 presents an example of this data file.
- o CEM/LOSS - This is the fifth and final file to be used as input by this utility. Using this data the utility will record in the output file, one record for each of the seven time periods in the study detailing within each record percentage losses per 30 days of the 12 types of equipment considered in the theater model. Figure III.18.8 presents an example of this data file.

Using these five input files the utility simply formats and writes the CONTROL/XX file as its output.

**18.1.2.B OUTPUT DATA AND DATA FILES:** There is only one output file produced by this utility. It is called the CONTROL/XX file. The file is cataloged as an element under the current study's program file and will be used as one of the three input to the ELCON utility. The file uses the exact same format as the CONTROL/TEMP file that was used as one of the utility's input files. Further, much of the data that was present in that file was simply copied to this output file as was discussed earlier. The remaining information contained in this file was supplied by the other four input files. It should be emphasized that no data in this file is derived; the data is simply read from the input files, organized and written to this output file by the utility.

The output from this utility will consist of eleven record types. The first format of the first record type will allow the user to control how the ELCON program will execute and how its output will be formatted and stored. The subsequent ten records will provide essential elements of information to the utility. A short description of these records and their contents are presented in Figure 3.16.9. Figure 3.16.10 presents an example of the data found in the file. Volume I of the documentation.

### 18.2.C VARIABLE DICTIONARY:

The following section identifies and defines the variables used by this utility.

<u>VARIABLE NAME</u>	<u>DEFINITION</u>
COPY	This is a 6 character alphanumeric variable which is dimensioned as an array with 12 occurrences by the utility. Data is read from the CONTROL/TEMP input file in records of 72 bytes into this array. The records are then immediately written to the output file CONTROL/COMPILER
THE	This is a 3 character alphanumeric variable which is used to locate and position the utility to the proper record within the input file CONTROL/DIVISIONS. The utility will read past records from the input file until this variable equals its namesake, i.e. "THE".
DIVS	This is an array within the utility dimensioned at 7 occurrences; one occurrence for each of the 7 time periods used within the study. Data found in this array reflect the average number of divisions in theater by time period. This data is read in from the input file COUNT/DIVISIONS.
ATTACK	This is a six character alphanumeric variable which is used as a check by the utility, similar to the discussed above. It allows the utility read past unneeded data in the input file SCENARIO/XX.
DUMMY	A six character alphanumeric variable that is used as an array within the utility. As its name implies it is simply a DUMMY variable used to read past unwanted data in the SCENARIO/XX input file.
SCEN	A 4 member array used by the utility to store the fraction of time spent by U.S. forces in each of the four combat postures. The sum of these four fractions must equal 1. This data is read from the input file SCENARIO/XX.
TIME1 - TIME7	These seven variables are used as arrays by the utility and are each dimensioned with enough occurrences to hold data for the number of theater cycles in each time period. Each theater cycle is equal to 4 days. TIME1 for example, reflects the first time period which has 15 days. Thus TIME1 is dimensioned at 4. TIME6 reflects a 30 day period and has 8 occurrences. Data found in these arrays is read from the input file REDARTY/DEGR-XX.

## TOTAL

This variable is used as an array by the utility. The array has seven occurrences; one occurrence for each of the seven time periods in the study. The value found in each occurrence of the array is calculated using the Red Artillery loss figures contained in the arrays TIME1 through TIME7. Supplied by the REDARTY/DEGR file using the following schemes:

$$\text{TOTAL}(1) = \text{TIME1}(1) = \text{TIME1}(2) + \text{TIME1}(3) + (\text{TIME1}(4) \times .75)$$

$$\text{TOTAL}(2) = (\text{TIME1}(4) \times .25) + (\text{TIME2}(4) \times .50) + \text{TIME2}(1) + \text{TIME2}(2) + \text{TIME2}(3)$$

$$\text{TOTAL}(3) = (\text{TIME2}(4) \times .50) + \text{TIME3}(1) + \text{TIME3}(2) + \text{TIME3}(3) + \text{TIME3}(4) + \text{TIME3}(5) + \text{TIME3}(6) + \text{TIME3}(7)$$

$$\text{TOTAL}(4) = \text{TIME4}(1) + \text{TIME4}(2) + \text{TIME4}(3) + \text{TIME4}(4) + \text{TIME4}(5) + \text{TIME4}(6) + \text{TIME4}(7) + (\text{TIME4}(8) \times .50)$$

$$\text{TOTAL}(5) = \text{TIME5}(1) + \text{TIME5}(2) + \text{TIME5}(3) + \text{TIME5}(4) + \text{TIME5}(5) + \text{TIME5}(6) + \text{TIME6}(7) + (\text{TIME4}(8) \times .50)$$

$$\text{TOTAL}(6) = \text{TIME6}(1) + \text{TIME6}(2) + \text{TIME6}(3) + \text{TIME6}(4) + \text{TIME6}(5) + \text{TIME6}(6) + \text{TIME6}(7) + (\text{TIME6}(8) \times .50)$$

$$\text{TOTAL}(7) = \text{TIME7}(1) + \text{TIME7}(2) + \text{TIME7}(3) + \text{TIME7}(4) + \text{TIME7}(5) + \text{TIME7}(6) + \text{TIME7}(7) + (\text{TIME6}(8) \times .50)$$

The next fraction in the utility is to determine the daily ammo expenditure or loss rates. This is done by dividing the seven time period totals by the number of days in the time periods, i.e. periods 1 and 2 = 15 days and periods 3-7 = 30 days.

## TEMP1

This is a temporary variable used in calculating the daily ammo expenditure rates.

## TEMP2

This variable is used to temporarily hold the values of the TOTAL array in order that comparisons of TOTAL can be made to determine the largest ammo expenditure period.

## FACT

This is an array of seven occurrences; one for each of the seven time periods in the study. This array is used to hold the resulting fraction of the division of the seven members in the total array by the TEMP2 variable, which contains the value of the highest TOTAL value. The resulting values will reflect the Red Artillery degradation factors for each of the seven time periods.

This variable is used as an array by the utility with 18 occurrences. The contents of the array will contain the average 15 day (for the first 2 periods) and 30 day (for the remaining periods) for the MIE Weapons identified as being played in the study. As currently structured for each period only 13 weapons types may be played. If more are to be played the number of fields the utility reads from the CEM/LOSSES input file must be increased from 13 to the required number. Further, if the number of weapon types exceeds 18, the dimension on the variable CLOSS must be increased.

KK,M,I,N,

Various integer variables used as subscripts and looping limitations within the utility.

NN,J,L,LL,  
MM,IJ,IK,IL,  
IM,IN,JI,JK,  
JL,JM,JN,KI,  
KJ,KL,K,LI,  
LJ,LK,LN,MI,  
MN,NJ,NM

**18.2 OPERATING ENVIRONMENT:** This program is implemented on the EXECUTIVE-8 operating system.

**18.2.1 SUPPORT SOFTWARE:** The utility requires the FORTRAN IV compiler. In addition, the utility requires the facilities of the UNIVAC 1100/82 system.

**18.2.2 I/O DEVICES:** - This utility uses five input files which reside on disk and produces an output file which will be maintained on disk.

**18.3 MAINTENANCE PROCEDURES:** This program is maintained by the MPP analyst.

**18.3.1 PROGRAMMING CONVENTIONS:** This utility follows FORTRAN programming conventions.

**18.3.2 INTERNAL ERROR ROUTINES:** - There are no explicitly written code to detect and handle error conditions within the utility. Only errors detected by the system will be identified.

# CONTROL/COMPILER STRUCTURE

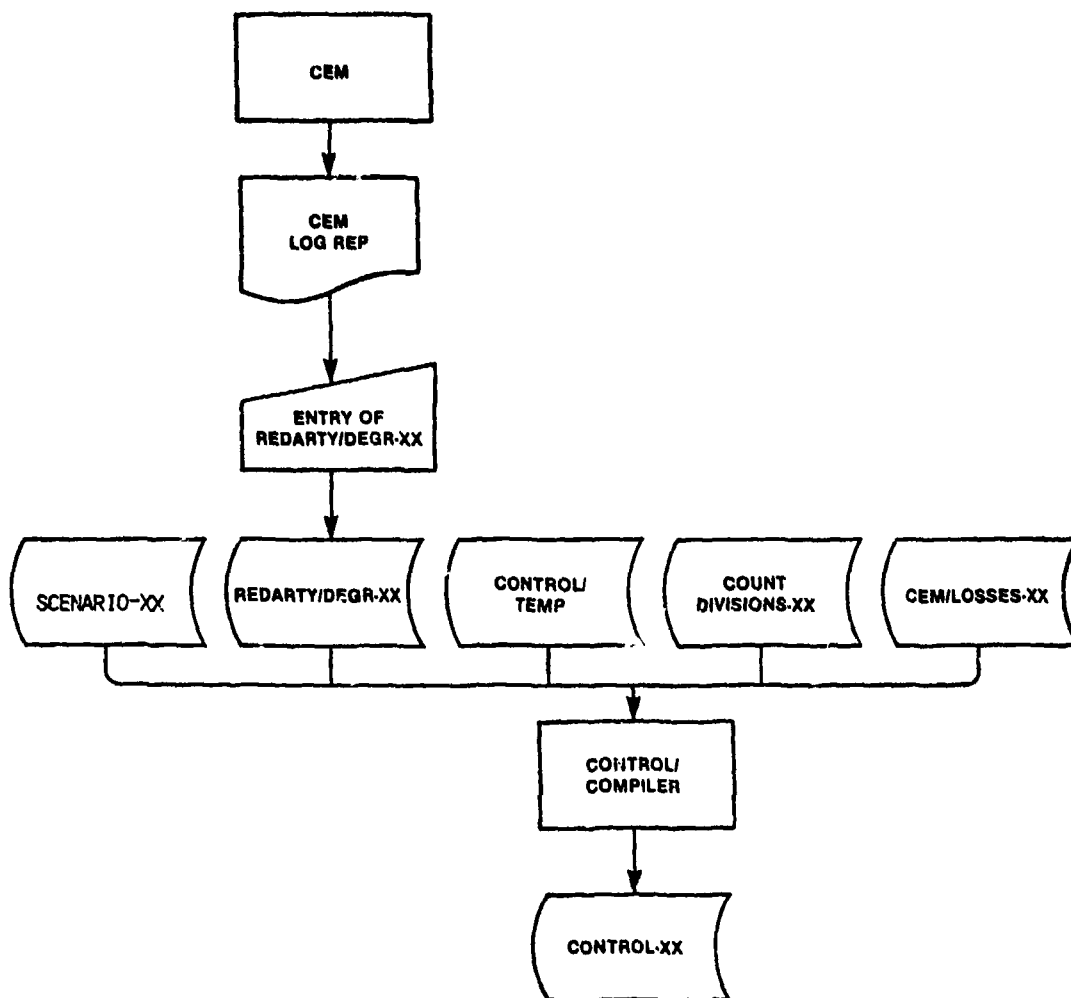


Figure III.18.1

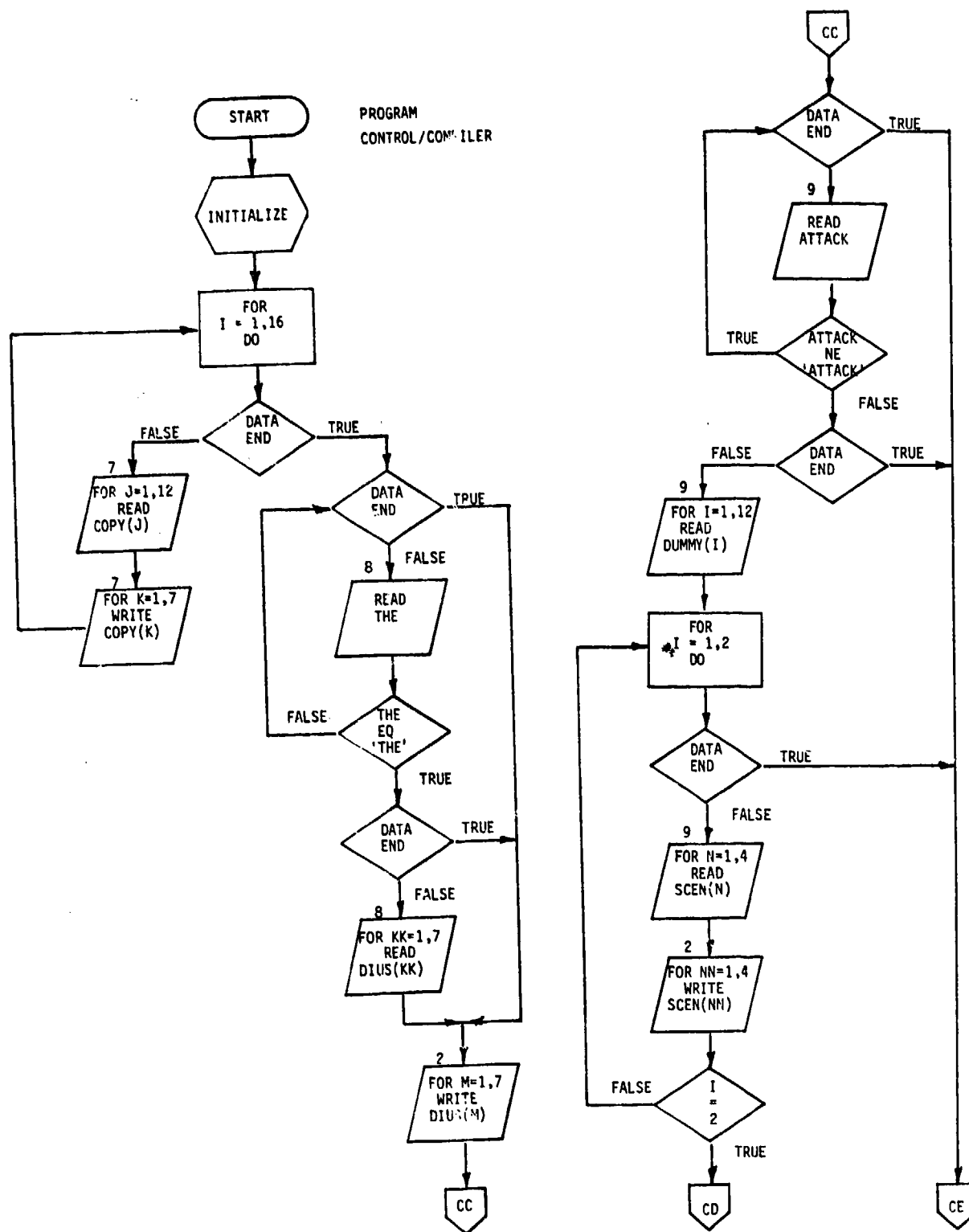


Figure III.18.2

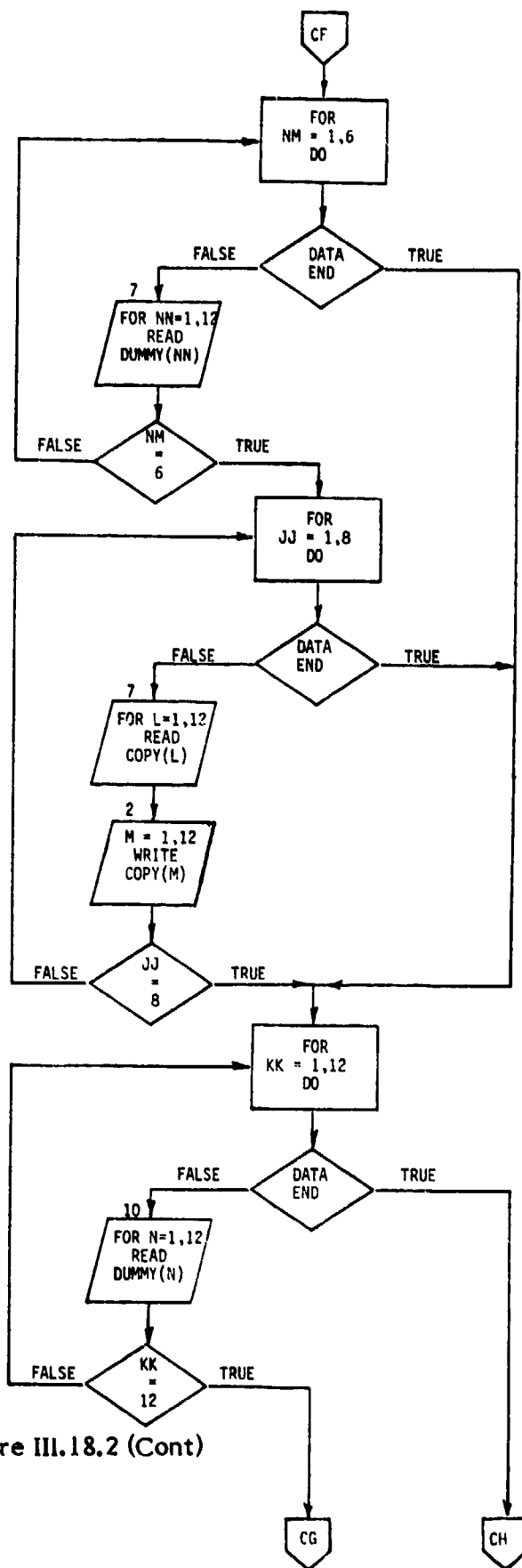
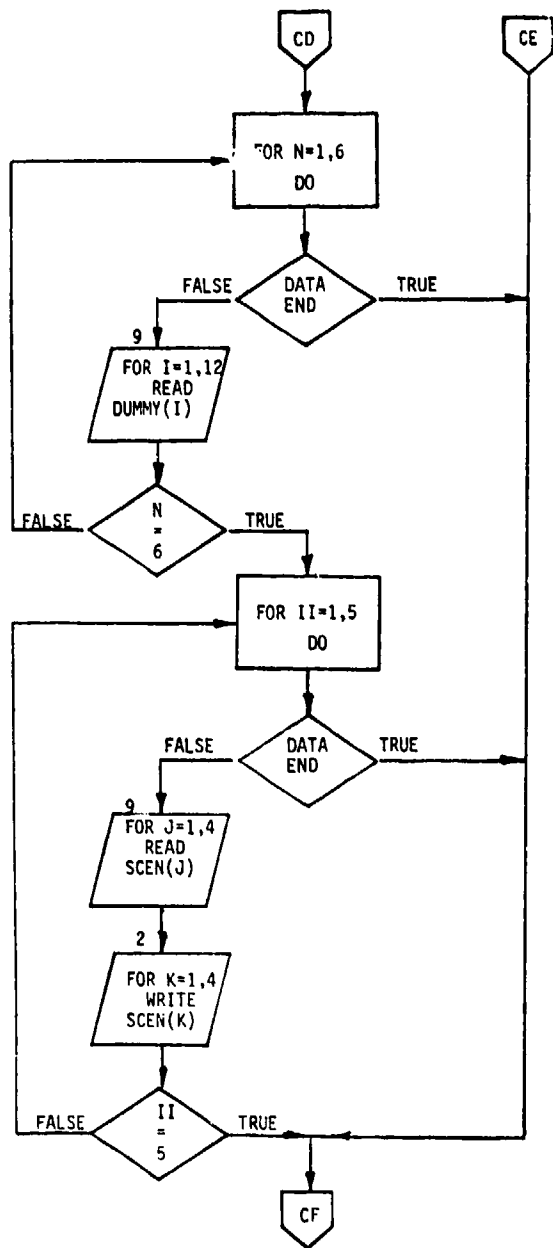
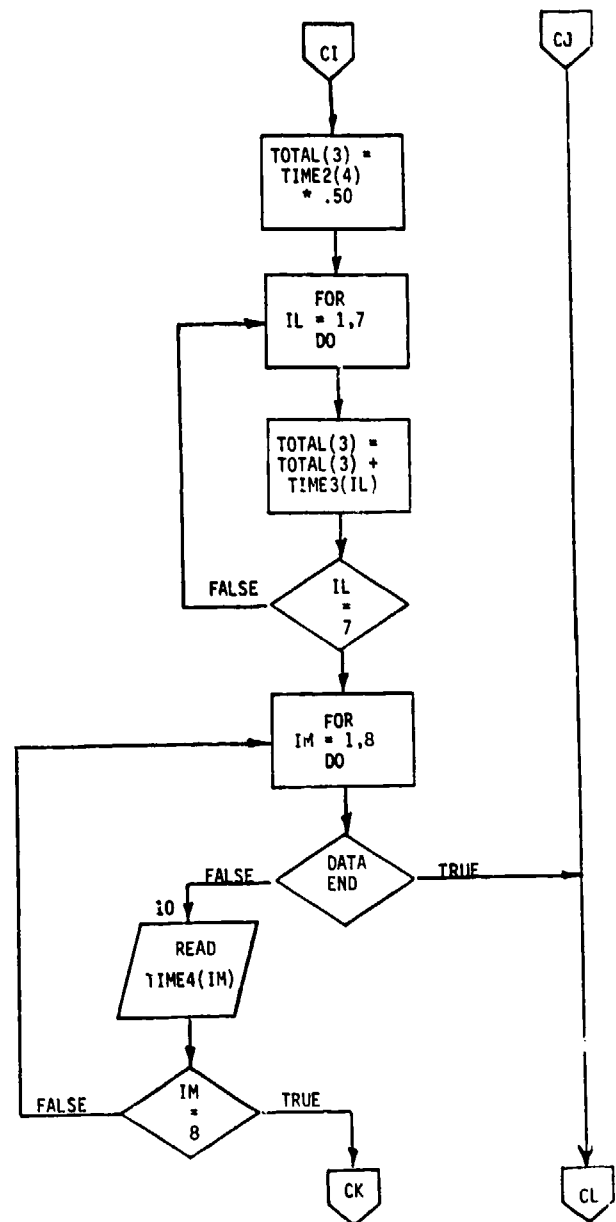
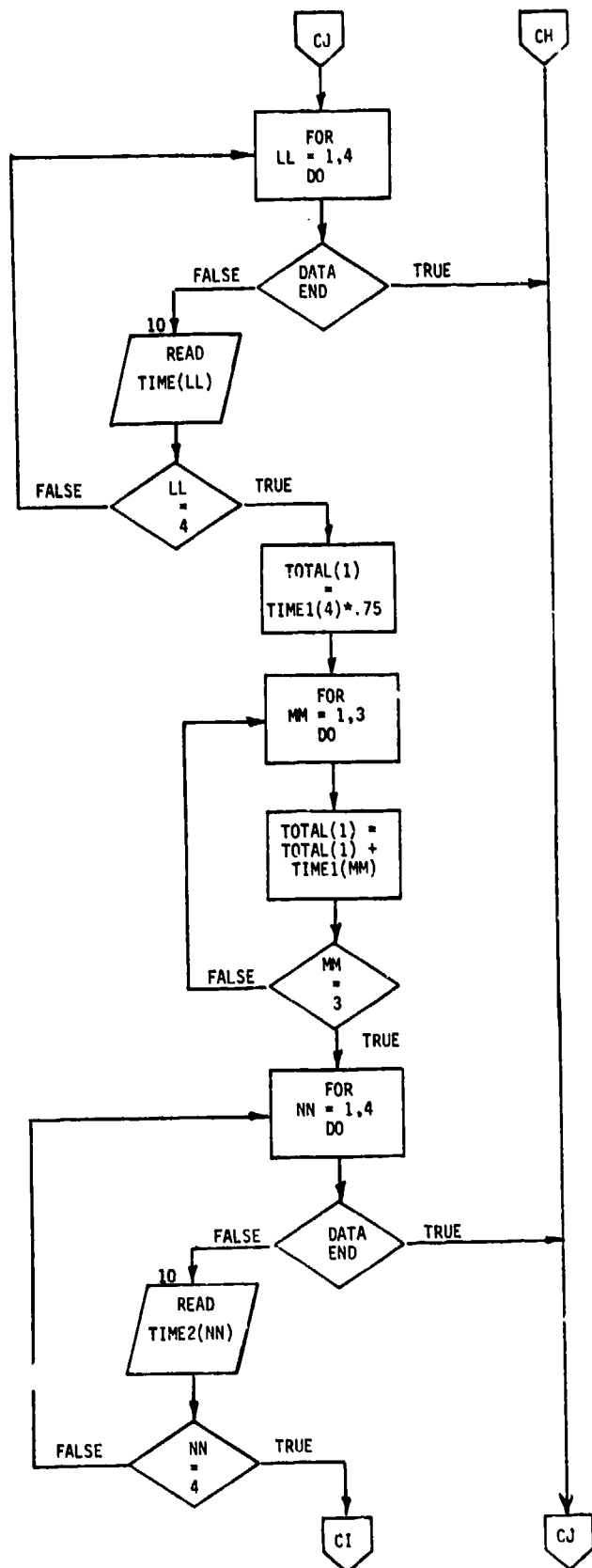


Figure III.18.2 (Cont)



(Cont)

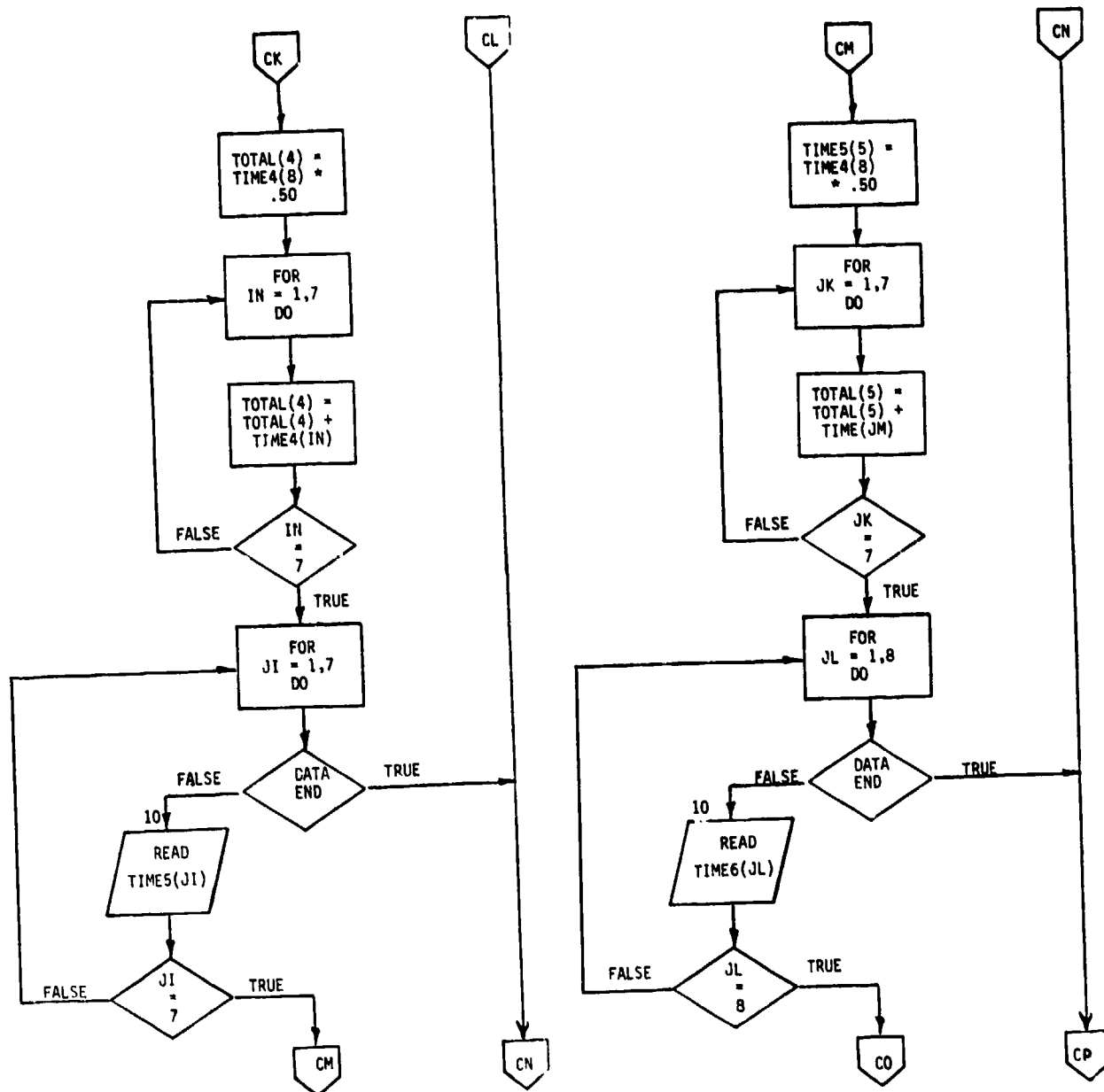


Figure III.18.2 (Cont)

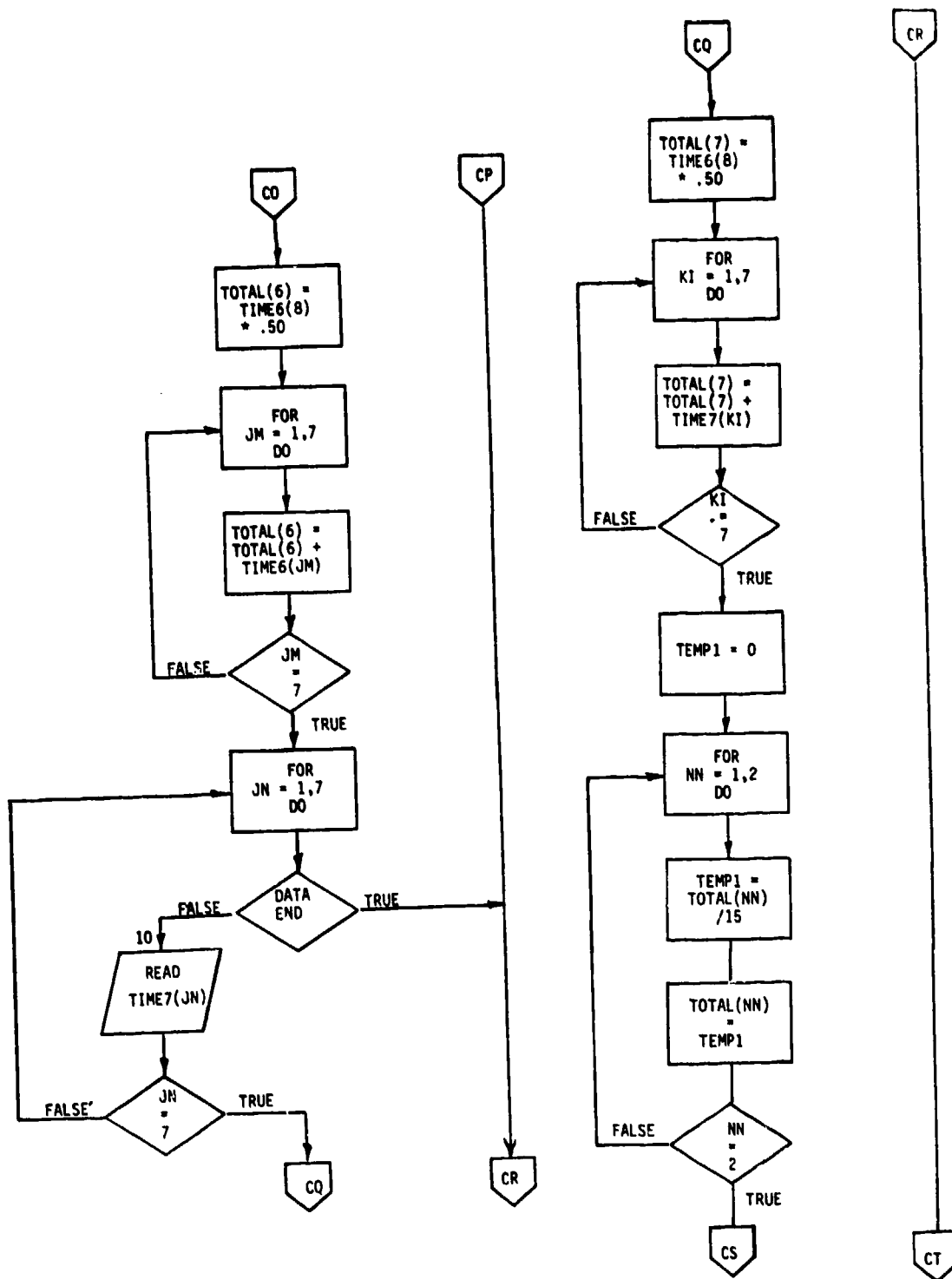
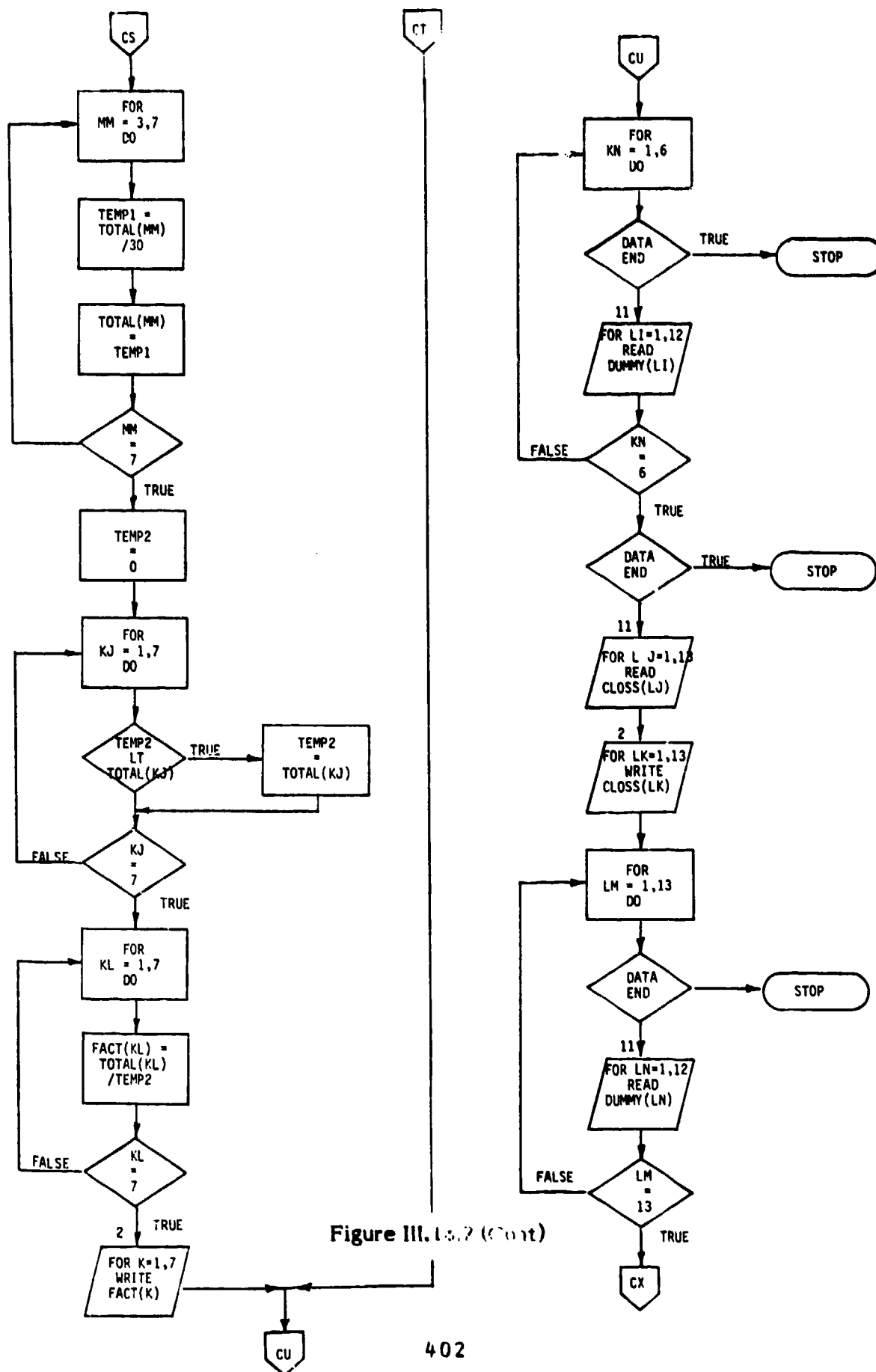


Figure III.18.2 (Cont)



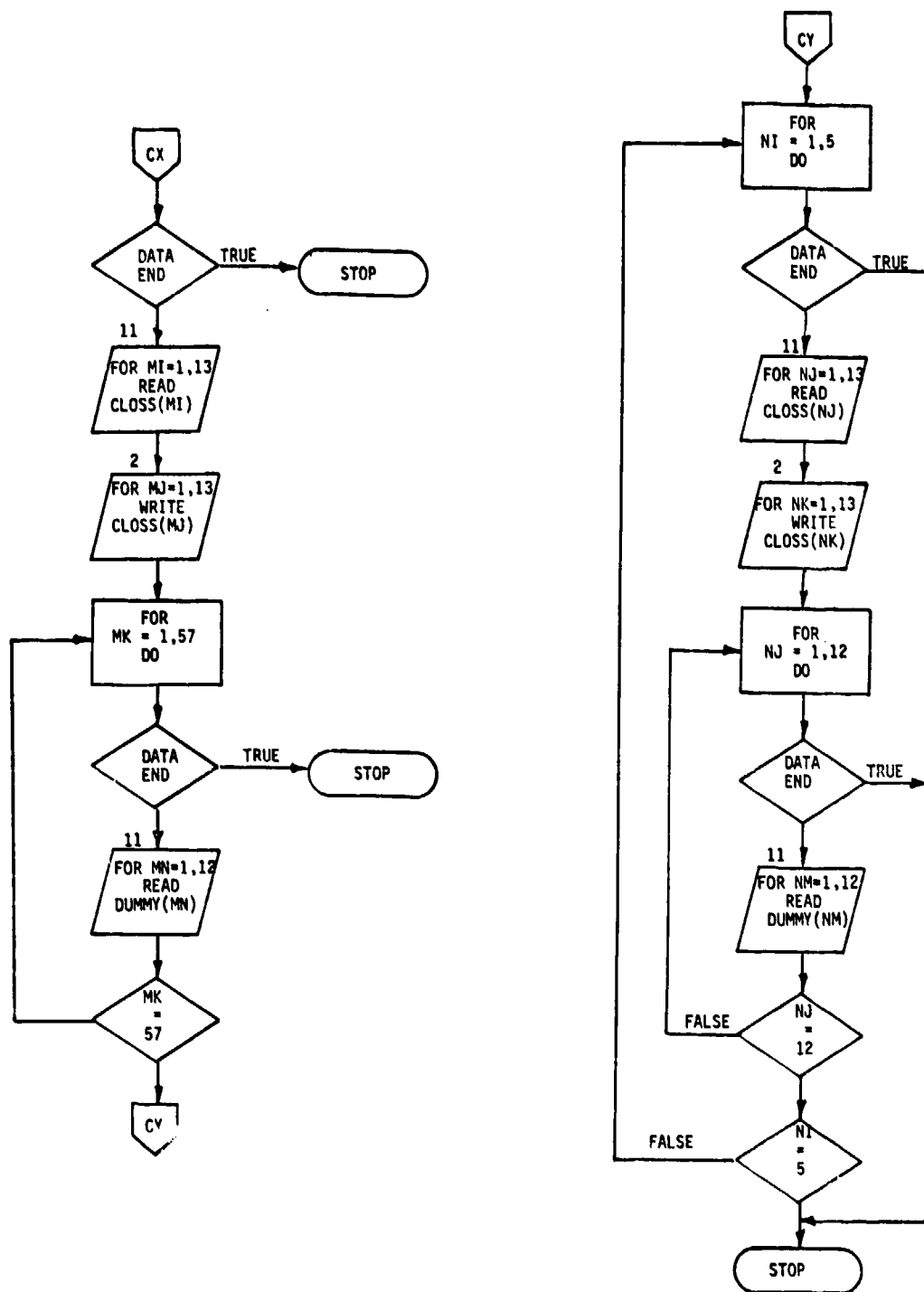


Figure III.18.2 (Cont)

**THIS PAGE IS INTENTIONALLY LEFT BLANK**

**402B**

UNCLASSIFIED\*\*FILE NAME:62XQT ELEMENT NAME:CONTROL/COMPILE\*\*UNCLASSIFI

```

1:C*****
2:C*****
3:C*****THIS PROGRAM IS WRITTEN IN FORTRAN IV.
4:C*****
5:C*****
6:C*****PURPOSE OF PROGRAM:
7:C*****
8:C*****THIS UTILITY IS DESIGNED TO COLLECT DATA FROM THREE
9:C*****UTILITIES (BASED ON CEM INPUT AND OUTPUT) AND TWO WARF ELEMENT (DATA)
10:C*****FILES. THE UTILITY USES THIS DATA TO COMPUTE VARIOUS CALCULATIONS
11:C*****AND CREATE THE CONTROL DATA FILE, WHICH IS STORED AS AN ELEMENT OF
12:C*****THE WARF STUDY'S PROGRAM FILE. THE CONTROL ELEMENT FILE WILL
13:C*****USE AS INPUT TO THE ELCON.
14:C*****THE FOLLOWING UTILITIES PROVIDE INPUT DATA
15:C*****AND MUST BE EXECUTED PRIOR TO THIS UTILITY:
16:C*****      1. 82XQT.SFARCH/ENGAGEREP
17:C*****      2. 82XQT.CFM/DATA
18:C*****      3. 82XQT.CFM/LOSSES
19:C*****      4. 82XQT.COUNT/DIVISIONS
20:C*****THE FOLLOWING ELEMENTS (DATA FILES) OF THE WARF STUDY'S
21:C*****PROGRAM FILE MUST BE CREATED PRIOR THE EXECUTION
22:C*****OF THIS UTILITY:
23:C*****      1. CONTROL/TEMP
24:C*****      2. RECDARTY/DEGR-XX
25:C*****
26:C*****
27:C*****VARIABLE DICTIONARY
28:C*****VARIABLE NAME      DEFINITION
29:C*****COPY(1)           HOLDS COPIED DATA FROM TEMPORARY CONTROL FILE.
30:C*****THE               USED TO VERIFY READ POSITION IN 82PRTS DATA
31:C*****                  FILE.
32:C*****DIVS(1)           CONTAINS NUMBER OF DIVISIONS PER TIME PERIOD.
33:C*****DUMMY(1)          USED TO READ PAST UNWANTED DATA.
34:C*****ATTACK            USED TO VERIFY POSITION IN DATA FILE 82SCENARIO.
35:C*****SCEN(1)           PERCENT OF TIME IN EACH POSTURE(I.E.ATTACK).
36:C*****TIME(1)-TIME(7)  AMOUNT OF AMMO EXPENDITURES FOR EACH FOUR DAY
37:C*****                  THEATER CYCLE FOR EACH TIME PERIOD(1-7).
38:C*****TOTAL(1)          TOTAL AMMO EXPENDITURE PER TIME PERIOD(1-7).
39:C*****FACT(1)           RESULTING RECDARTY DEGRADATION FACTOR FOR EACH
40:C*****                  TIME PERIOD(1-7).
41:C*****TEMP              USED TO FIND AND HOLD HIGHEST AMMO EXPENDITURE
42:C*****                  OF THE SEVEN TIME PERIODS.
43:C*****CLOSS(1)          USED TO STORE THE CEM LOSS RATE PERCENTS.
44:C*****
45:C*****
46:      DIMENSION COPY(12),DUMMY(12),DIVS(7),TIME(14),TIME2(14),TIME3(7),
47:      *TIME4(8),TIME5(7),TIME6(6),TIME7(7),TOTAL(7),FACT(7),CLOSS(15),
48:      *SCEN(4)
49:C*****
50:C*****THIS SECTION COLLECTS AND RECOPIES DATA FROM THE TEMPORARY CONTROL
51:C*****FILE TO THE FILE CONTROL FILE.
52:C*****
53:      DO 10 J=1,16
54:      READ(7,100,END=99)(COPY(J),J=1,12)
55:100  FORMAT(12A6)
56:      WRITE(10,101)(COPY(K),K=1,7)
57:10  CONTINUE

```

Figure III.18.3

UNCLASSIFIED\*\*\*FILE NAME:BPXCT ELEMENT NAME:CONTROL/COMPILE\*\*\*UNCLASSIFI

```

68:C*****
69:C*****THIS SECTION READS DATA FROM PROVISIONS DATA FILE FROM BPXCT.COUNTDIVS.
70:C*****
61:99      READ(9,101,END=98)THL
62:101     FORMAT(1X,A7)
63:        IF(THL.EQ.'THE')GO TO 7P
64:        GO TO 99
65:78      READ(9,103,END=98)(DIVS(KK),KK=1,7)
66:103     FORMAT(12A6)
67:103     FORMAT(1X,7(F4.1,1X))
68:C*****
69:C*****RITE DIVISIONS COUNT IN THE FINAL CONTROL FILE.
70:C*****
71:98      WRITE(2,201)(DIVS(M),M=1,7)
72:201     FORMAT(7(1X,F4.1))
73:C*****
74:C*****READING IN DATA FROM 92SCENARIO ON PERCENT OF TIME IN EACH POSTURE.
75:C*****
76:13      READ(9,104,END=97)ATTACK
77:104     FORMAT(5X,A6)
78:        IF(ATTACK.NE.'ATTACK')GO TO 13
79:        READ(9,102,END=97)(DUMMY(I),I=1,12)
80:        DO 14 I=1,2
81:        READ(9,105,END=97)(SCEN(N),N=1,4)
82:105     FORMAT(5X,4(F6.4,4X))
83:        WRITE(2,202)(SCEN(NN),NN=1,4)
84:202     FORMAT(4F5.7)
85:14      CONTINUE
86:        DO 15 N=1,6
87:        READ(9,102,END=97)(DUMMY(I),I=1,12)
88:15      CONTINUE
89:        DO 16 J=1,5
90:        READ(9,105,END=97)(SCEN(J),J=1,4)
91:        WRITE(2,202)(SCEN(K),K=1,4)
92:16      CONTINUE
93:C*****
94:C*****COPYING AND WRITING WARFAM LOSSES FROM THE TEMPORARY TO THE FINAL CONTROL
95:C      FILE.
96:C*****
97:97      DO 117 N=1,6
98:        READ(7,102,END=96)(DUMMY(NN),NN=1,12)
99:117     CONTINUE
100:        DO 17 JJ=1,5
101:        READ(7,102,END=96)(COPY(L),L=1,12)
102:        WRITE(2,102)(COPY(M),M=1,12)
103:17      CONTINUE
104:C*****
105:C*****COMPUTATIONS ON AMMO DATA FROM SCHWARFSYS.REDARTY/DEGR
106:C      TO OBTAIN THE RED ARTY DEGRADATION FACTOR.
107:C*****
108:96      DO 18 KK=1,12
109:        READ(10,102,END=95)(DUMMY(N),N=1,12)
110:18      CONTINUE
111:        DO 19 LL=1,4
112:        READ(10,106,END=95)TIME(LLL)
113:19      FORMAT(112X,F7.0)
114:19      CONTINUE

```

Figure III.18.3 (Cont)

UNCLASSIFIED\*\*\*FILE NAME:182RCT ELEMENT NAME:CONTROL/COMPILER\*\*\*UNCLASSIFIED

```

115:      TOTAL(1)=TIME(14)*.75
116:      DO 20 MM=1,7
117:      TOTAL(1)=TOTAL(1)+TIME1(MM)
118:20      CONTINUE
119:      DO 21 NN=1,4
120:      READ(10,106,END=95)TIME2(NN)
121:21      CONTINUE
122:      TOTAL(2)=(TIME1(4)*.25)+(TIME2(4)*.50)
123:      DO 22 IJ=1,7
124:      TOTAL(2)=TOTAL(2)+TIME2(IJ)
125:22      CONTINUE
126:      DO 23 IK=1,7
127:      READ(10,106,END=95)TIME3(IK)
128:23      CONTINUE
129:      TOTAL(3)=TIME2(4)*.50
130:      DO 24 IL=1,7
131:      TOTAL(3)=TOTAL(3)+TIME3(IL)
132:24      CONTINUE
133:      DO 25 IM=1,5
134:      READ(10,106,END=95)TIME4(IM)
135:25      CONTINUE
136:      TOTAL(4)=TIME4(5)*.50
137:      DO 26 IN=1,7
138:      TOTAL(4)=TOTAL(4)+TIME4(IN)
139:26      CONTINUE
140:      DO 27 OI=1,7
141:      READ(10,106,END=95)TIME5(OI)
142:27      CONTINUE
143:      TOTAL(5)=TIME4(5)*.50
144:      DO 28 OJ=1,7
145:      TOTAL(5)=TOTAL(5)+TIME5(OJ)
146:28      CONTINUE
147:      DO 29 OL=1,4
148:      READ(10,106,END=95)TIME6(OL)
149:29      CONTINUE
150:      TOTAL(6)=TIME5(5)*.50
151:      DO 30 OM=1,7
152:      TOTAL(6)=TOTAL(6)+TIME6(OM)
153:30      CONTINUE
154:      DO 31 ON=1,7
155:      READ(10,106,END=95)TIME7(ON)
156:31      CONTINUE
157:      TOTAL(7)=TIME6(5)*.50
158:      DO 32 KI=1,7
159:      TOTAL(7)=TOTAL(7)+TIME7(KI)
160:32      CONTINUE
161:      TEMP1=0
162:      DO 41 NN=1,7
163:      TEMP1=TOTAL(INN)/15
164:      TOTAL(INN)=TEMP1
165:41      CONTINUE
166:      DO 42 MM=1,7
167:      TEMP1=TOTAL(MM)/30
168:      TOTAL(MM)=TEMP1
169:42      CONTINUE
170:      TEMP2=0
171:      DO 33 MJ=1,7

```

Figure III.18.3 (Cont)

UNCLASSIFIED\*\*\*FILE NAME:12X01 ELEMENT NAME:CONTROL/COMPILER\*\*\*UNCLASSIFIED

```

172:      IF(TEMP2.LT.TOTAL(KJ))TEMP2=TOTAL(KJ)
173:33      CONTINUE
174:      DO 34 KL=1,7
175:          FACT(KL)=TOTAL(KL)/TEMP2
176:34      CONTINUE
177:      WRITE(2,221)(FACT(K),K=1,7)
178:221      FORMAT(1X,7F5.3)
179:C*****
180:C*****READING FROM 82CEMLOSS2 CFM LOSSES AND WRITING INTO FINAL CONTROL FILE.
181:C*****
182:95      DO 35 KN=1,6
183:      READ(11,102,END=94)(DUMMY(LI),LI=1,12)
184:35      CONTINUE
185:      READ(11,107,END=94)(CLOSS(LJ),LJ=1,13)
186:107      FORMAT(15X,15F7.1)
187:      WRITE(2,204)(CLOSS(LK),LK=1,17)
188:204      FORMAT(1X,17F6.2)
189:      DO 36 LM=1,13
190:      READ(11,102,END=94)(DUMMY(LN),LN=1,12)
191:36      CONTINUE
192:      READ(11,107,END=94)(CLOSS(MI),MI=1,13)
193:      WRITE(2,204)(CLOSS(PJ),PJ=1,13)
194:      DO 37 PK=1,57
195:      READ(11,102,END=94)(DUMMY(MN),MN=1,12)
196:37      CONTINUE
197:      DO 38 NI=1,5
198:      READ(11,107,END=94)(CLOSS(NJ),NJ=1,13)
199:      WRITE(2,204)(CLOSS(NK),NK=1,17)
200:      DO 39 NJ=1,12
201:      READ(11,102,END=94)(DUMMY(NM),NM=1,12)
202:39      CONTINUE
203:94      STOP
204:      END

```

Figure III.18.3 (Cont)

UNCLASSIFIED\*\*\*EXAMPLE OF A CONTROL/TEMP DATA FILE\*\*\*UNCLASSIFIED

```

1:ADD 88.CONTROL/TEMP
2:TEMPORARY CONTROL DATA FILE WARF P88(12DEC80)
3: 7 1200 13 10 0 2 -1
4: 15 15 30 30 30 30 30
5: 1 1
6: 2 2
7: 1 2
8: 3 3
9: 4 4
10: 5 5
11: 6 6
12: 7 7
13: 1 4
14: 5 7
15: .05 .05 .05 .01 .01 .00 .00
16: .15 .15 .23 .10 .05 .02 .00
17: .15 .15 .10 .10 .05 .05 .05
18: 0.0 0.0 0.0 0.0 0.0 0.0 0.0
19: .000 .000 .000 .000
20: .000 .000 .000 .000
21: .000 .000 .000 .000
22: .000 .000 .000 .000
23: .000 .000 .000 .000
24: .000 .000 .000 .000
25: .000 .000 .000 .000
26: .000 .994 .211 1.666 .222 .111 .666 .090 .138 .899 .574
27: .224 .691 .264 .714 .999 .744 .359 .907 .644 .921 .100
28: .100 .992 .600 2.744 .578 .000 .299 .000 .704 .000 2.192
29: 7.682 3.339 .668 1.750 1.199 .502 .078 .515 .597 .000 .080
30: .200 2.507 1.254 .927 .494 .000 .633 .000 .283 .284 1.609
31: 6.753 3.355 .895 3.001 1.678 1.789 .465 .898 .566 .000 .090
32: .050 .255 .064 .555 .888 .010 .088 .550 .823 .760 .064
33: 3.142 7.460 .533 .850 .973 .349 .077 .050 .778 .000 .080
34: .000 .000 .000 .000 .000 .000 .000
35: .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00
36: .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00
37: .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00
38: .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00
39: .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00
40: .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00
41: .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00
42: .....
43:COMMENTS:
44: THE ABOVE ELEMENT FILE "CONTROL/TEMP" IS CREATED BY MANUALLY
45: EDITTING A PREVIOUS STUDIES CONTROL FILE INTO THE CURRENT
46: WARF STUDY'S PROGRAM FILE (IN THIS EXAMPLE "SECRET*82WARFP88").
47: LINES 1-16 MUST BE EDITTED WITH THE CURRENT STUDY DATA ON
48: REPORTING AS DIRECTED BY THE STUDY DIRECTOR AND IAW TABLES
49: 4-4 AND 4-5 OF THE ELCON MANUAL CAA-D-79-3, DATED AUGUST 1979.
50: THESE LINES CONTAIN DATA ON THE #TIME PERIODS, #LIN CODES,
51: #CEM ITEMS, #RATES TO BE COMPUTED, #DAYS OF PREPOSITIONED
52: STOCK, TYPE OF REPORT, LENGTH OF TIME PERIODS, INTERTHEATER
53: LOGISTIC (SEA AND AIR) LOSSES, AND INTRATHEATER LOGISTIC
54: LOSSES. FOLLOWING THE COMPLETION OF WARFRAM AND THE CALCUL-
55: ATION OF LOSS RATES FOR THE 22 ARTILLERY VULNERABILITY
56: CATEGORIES FOR EACH COMBAT POSTURE LINES 25-32 MUST BE FILLED
57: IN WITH THIS DATA IAW TABLE 4-5 OF THE ELCON MANUAL MENTIONED
58: ABOVE. THESE LINE NUMBERS MAY CHANGE DUE TO THE ADDITION OR
59: DELETION OF DATA. THE REMAINDER OF THE FILE CURRENTLY SHOWN
60: WITH ZEROS WILL BE FILLED OUT BY THE EXECUTION OF UTILITY
61: "82XQT.CONTROL/COMPILER". THIS UTILITY WILL CREAT A NEW
62: CONTROL FILE ELEMENT, WHICH WILL BE USED IN THE EXECUTION
63: OF ELCON.

```

Figure III.18.4

UNCLASSIFIED\*\*\*EXAMPLE OF THE OUTPUT FROM UTILITY COUNT/DIVISIONS\*\*\*UNCLASSIFIED

```
1: TOTAL NUMBER OF ARRIVING DIVISIONS BY PERIOD.  
2: 8 2 0 0 0 0  
3: AVERAGE NUMBER OF ARRIVING DIVISIONS BY PERIOD.  
4: 6.062 1.500 0.000 0.000 0.000 0.000 0.000  
5: NUMBER OF US DIVISIONS IN THEATER ON D-DAY.  
6: 42  
7: THE AVERAGE DIVISIONAL COUNT BY PERIOD.  
8: 48.1 51.5 52.0 52.0 52.0 52.0 52.0
```

Figure III.18.5

UNCLASSIFIED\*\*\*EXAMPLE OF THE OUTPUT FROM UTILITY SEARCH/ENGAGEPP\*\*\*UNCLASSIFIED

1:	IC YCLE= 1	BLUE PARTITION 1	
2:	IC YCLE= 2	BLUE PARTITION 1	
3:	IC YCLE= 3	BLUE PARTITION 1	
4:	IC YCLE= 4	BLUE PARTITION 1	
5:	IC YCLE= 5	BLUE PARTITION 1	
6:	IC YCLE= 6	BLUE PARTITION 1	
7:	IC YCLE= 7	BLUE PARTITION 1	
8:	IC YCLE= 8	BLUE PARTITION 1	
9:	IC YCLE= 9	BLUE PARTITION 1	
10:	IC YCLE= 10	BLUE PARTITION 1	
11:	IC YCLE= 11	BLUE PARTITION 1	
12:	IC YCLE= 12	BLUE PARTITION 1	
13:	IC YCLE= 13	BLUE PARTITION 1	
14:	IC YCLE= 14	BLUE PARTITION 1	
15:	IC YCLE= 15	BLUE PARTITION 1	
16:		.0	
17:		.0	
18:		.0	
19:		.0	
20:		243.7	
21:		338.2	
22:		39.6	
23:		2570.6	
24:		202.0	
25:			.000
26:			.171
27:			.012
28:			.817
29:	IC YCLE= 16	BLUE PARTITION 1	
30:	IC YCLE= 17	BLUE PARTITION 1	
31:	IC YCLE= 18	BLUE PARTITION 1	
32:	IC YCLE= 19	BLUE PARTITION 1	
33:	IC YCLE= 20	BLUE PARTITION 1	
34:	IC YCLE= 21	BLUE PARTITION 1	
35:	IC YCLE= 22	BLUE PARTITION 1	
36:	IC YCLE= 23	BLUE PARTITION 1	
37:	IC YCLE= 24	BLUE PARTITION 1	
38:	IC YCLE= 25	BLUE PARTITION 1	
39:	IC YCLE= 26	BLUE PARTITION 1	
40:	IC YCLE= 27	BLUE PARTITION 1	
41:	IC YCLE= 28	BLUE PARTITION 1	
42:	IC YCLE= 29	BLUE PARTITION 1	
43:	IC YCLE= 30	BLUE PARTITION 1	
44:		213.9	
45:		339.5	
46:		286.3	
47:		30.3	
48:		475.9	
49:		644.0	
50:		70.7	
51:		5314.4	
52:		457.0	
53:			.109
54:			.145
55:			.009
56:			.737
57:	IC YCLE= 31	BLUE PARTITION 1	

Figure III.18.6

UNCLASSIFIED//EXEMPT OF INPUT DATA REDACTED/DIGR TO UTILITY CONTROL/COMPILER

CFM REPORT GENERATOR														
33CFM-RUN/PPH-PLS NAP DTGCTED UNCLASSIFIED														
AFR LOGISTIC EXPERIENCE BY MAJOR ITEM TYPE														
QUANTITIES SHOWN FOR EACH OF THREAT CYCLE UNLESS OTHERWISE LABELLED														
THREAT CYCLE	CATEGORY	APTHAM	TYPE WITHIN CATEGORY	TOTAL STOCKS	THREAT CYCLE	IN RE PAIR	COMBAT LOSS EXPERIENCE	TOTAL COMBAT	NON-COMBAT LOSSES	TOTAL LOSSES	THREAT CYCLE	THREAT CYCLE	THREAT CYCLE	THREAT CYCLE
10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
10	0	105775	105775	100.0	N/A	N/A	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A
11	1	117525	117525	99.7	110111	0	410111	410111	0	410111	0	0	0	410111
12	2	131425	131425	99.7	130111	0	130111	130111	0	130111	0	0	0	130111
13	3	110715	110715	99.7	110111	0	110111	110111	0	110111	0	0	0	110111
14	4	151915	151915	99.7	150111	0	150111	150111	0	150111	0	0	0	150111
15	5	171640	171640	99.7	170111	0	170111	170111	0	170111	0	0	0	170111
16	6	187440	187440	99.7	180111	0	180111	180111	0	180111	0	0	0	180111
17	7	191540	191540	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
18	8	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
19	9	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
20	10	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
21	11	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
22	12	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
23	13	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
24	14	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
25	15	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
26	16	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
27	17	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
28	18	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
29	19	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
30	20	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
31	21	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
32	22	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
33	23	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
34	24	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
35	25	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
36	26	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
37	27	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
38	28	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
39	29	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
40	30	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
41	31	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
42	32	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
43	33	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
44	34	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
45	35	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
46	36	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
47	37	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
48	38	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
49	39	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
50	40	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
51	41	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
52	42	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
53	43	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
54	44	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
55	45	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
56	46	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
57	47	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
58	48	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
59	49	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
60	50	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
61	51	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
62	52	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
63	53	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
64	54	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
65	55	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
66	56	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
67	57	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
68	58	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
69	59	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
70	60	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
71	61	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
72	62	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
73	63	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
74	64	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
75	65	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
76	66	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
77	67	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
78	68	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
79	69	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
80	70	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
81	71	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
82	72	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
83	73	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
84	74	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
85	75	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
86	76	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
87	77	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
88	78	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
89	79	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
90	80	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
91	81	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
92	82	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
93	83	197740	197740	99.7	190111	0	190111	190111	0	190111	0	0	0	190111
94	84													

UNCLASSIFIED\*\*\*EXAMPLE OF THE OUTPUT FROM UTILITY GCM/LOSSES\*\*\*UNCLASSIFIED

1: 15 1 15	AVERAGE 15-DAY LOSS RATES **D-1 TO D-15**									
2:										
3:	.12	.07	.05	12.79	58.56	4.65	34.47	125.66	58.95	7.15
4:	.12	.11	.18	13.17	1.75	7.97	13.10	126.64	225.05	217.27
5:	.36	3.32	55.97	.75	.39					
6:	.00	.00	.09	.00	92.62	147.30	326.03	25.58	.00	.00
7:R09CEM	01-9	64.0	14.0	.0	10.0	24.0	36.0	72.0	70.0	14.0
8:										
9:										
10:										
11:										
12:										
13:										
14:	15 16 30									
15:										
16:	.28	.07	.51	.10	5.42	9.98	34.50	102.00	70.93	76.13
17:	.17	.12	.26	7.53	2.06	8.23	89.24	54.41	147.71	110.92
18:	.40	12.65	8.70	.42	.71					
19:	.00	.00	.09	.05	33.72	54.12	105.59	11.38	.00	.00
20:R09CEM	01-9	56.0	14.0	122.0	70.0	34.0	52.0	60.0	42.0	18.0
21:										
22:										
23:										
24:										
25:										
26:										
27:	30 1 30									
28:	30 1 60									
29:	30 1 90									
30:										
31:	.46	.09	.50	.12	.00	.00	.00	.00	.00	.00
32:	.23	.15	.33	.00	.00	.00	.00	.00	.00	.00
33:	.27	.00	.00	.55	.27					
34:	.00	.00	.14	.07	.00	.00	.00	.00	.00	.00
35:R09CEM	01-9	40.0	9.0	50.0	12.0	23.0	33.0	27.0	27.0	14.0
36:										
37:										
38:										
39:										
40:										
41:										
42:	30 31 12									
43:	30 31 15									
44:	30 31 18									
45:	.72	.03	.32	.04	.00	.00	.00	.00	.00	.00
46:	.11	.03	.21	.00	.70	.00	.00	.00	.00	.00
47:	.03	.00	.00	.11	.03					
48:	.00	.00	.04	.00	.00	.00	.00	.00	.00	.00
49:	01-9	22.0	3.0	82.0	4.0	11.0	21.0	3.0	3.0	0.0
50:R09CEM	01-9	22.0	3.0	82.0	4.0	11.0	21.0	3.0	3.0	0.0

Figure III.18.8

UNCLASSIFIED\*\*\*EXAMPLE OF THE OUTPUT DATA FROM UTILITY CONTROL/COMPILE\*\*\*UNCLAS

```

1:TEMPORARY CONTROL DATA FILE NAME PROVIDED
2:  7      6      9      10     0      2     -1
3:  15     15     20     20     30     30     30
4:  1      1
5:  2      2
6:  1      2
7:  3      3
8:  4      4
9:  5      5
10: 6      6
11: 7      7
12: 1      4
13: 5      7
14: .05 .05 .05 .01 .01 .00 .00
15: .15 .15 .23 .10 .05 .02 .00
16: .15 .25 .10 .10 .05 .05 .05
17: 45.1 51.5 52.0 52.0 52.0 52.0 52.0
18: .000 .141 .059 .800
19: .255 .145 .050 .600
20: .200 .155 .005 .640
21: .100 .055 .105 .840
22: .005 .055 .140 .800
23: .705 .005 .000 .690
24: .555 .000 .000 .445
25: .000 .994 .711 1.660 .222 .111 .560 .090 .135 .999 .574
26: .024 .691 .264 .714 .999 .744 .353 .907 .644 .921 .100
27: .100 .992 .600 2.744 .579 .000 .299 .000 .704 .000 2.192
28: 7.582 3.339 .000 1.750 1.199 .502 .079 .515 .597 .000 .080
29: .200 2.507 1.254 .927 .494 .000 .633 .000 .287 .294 1.609
30: 0.753 3.355 .895 3.001 1.673 1.789 .465 .399 .565 .000 .090
31: .050 .255 .064 .555 .888 .010 .088 .550 .923 .760 .064
32: 3.142 7.460 .533 .850 .973 .349 .077 .050 .779 .000 .080
33: .649 .764 .980 1.000 .950 .485 .196
34: 64.00 1.40 .00 .00 10.00 24.00 35.00 72.00 7.30 .00 9.00 .00 .00
35: 50.00 1.40 .00 2.00 20.00 34.00 52.00 80.00 4.20 .00 3.00 .00 .00
36: 50.00 .90 .00 2.00 15.00 24.00 77.00 29.00 7.30 .00 7.00 .00 .00
37: 57.00 .50 .00 .00 6.00 18.00 25.00 10.00 .60 .00 .00 .00 .00
38: 19.00 .40 .00 .00 5.00 13.00 16.00 4.00 .50 .00 6.00 .00 .00
39: 30.00 .30 .00 4.00 4.00 14.00 33.00 3.00 .10 .00 4.00 .00 .00
40: 17.00 .10 .00 2.00 3.00 7.00 15.00 1.00 .10 .00 1.00 .00 .00

```

Figure III.18.9

## CHAPTER 19

### UTILITY - FINAL REPORT

19.1 DESCRIPTION OF PROCESSING: This program performs calculations in addition to the logic tested read and write statements.

19.1.1 PURPOSE/FUNCTIONS: The purpose of this utility is to combine the three output files from the ELCON program and the one ITMID/FINAL output file to produce a report on WARF daily attrition rates. Since the output from ELCON expresses loss rates in 30 day increments computations are performed to determine the daily rates. The only output from this utility is the printed report for the combat analyst. The report is not used as input to any other automated program.

19.1.2 PROGRAM INPUT/OUTPUT STRUCTURE: The overall structure of the utility is pictured in Figure III.19.1. The logic flow of the utility is depicted in Figure III.19.2. The source code of the utility is listed in Figure III.19.3.

19.1.2.A INPUT DATA AND DATA BASE: The utility uses four input files, all cataloged under the current study's program file, in this instance, SECRET\*82WARFP88. These files are the ITMID/FINAL, RATES-XX/SEC-1, RATES-XX/MONTHLY-WOL and RATES-XX/MONTHLY-WL. Each file is discussed below.

- o ITMID/FINAL - This file is produced by the ITMID-REC-A utility. It details each major item of equipment being analyzed in the study. For each item it provides the item's code identifier, nomenclature or description, its CEM type artillery vulnerability/historical class (where applicable) as well as a period by period summary of the authorized quantities of this item plus a combat zone by combat zone summary of the distribution or density of the item throughout the battle area. Figure 3.19.4 presents the file layout and example of the data for this file.
- o RATES-XX/SEC-1 - This file is produced by the ELCON program. It summarizes for the utility the control information under which the ELCON program was executed. In addition to a summary for each major piece of equipment a summary of its loss rates in different periods from a variety of causes. The USACAA document CAA-D-79-3 in Appendix B presents a complete example of a typical file on pages D1 - D12. Item Y of Appendix B also contains details on this file. The FINAL/REPORT program will simply read this file and write the contents out.

It should be noted that the ELCON program discussed in Chapter 2 has the option of executing without producing this file. If this is the case, the FINAL/REPORT program will not execute because it expects the file to exist. Figure III.19.5 presents an example of this data file.

- o RATES-XX/MONTHLY-WOL - This file details for each major item of

equipment in the study in-theater monthly losses per WARF set excluding LOC, depot and inter-theater logistic losses. Figure III.19.6 presents the file layout and sample data for this file.

- o RATES-XX/MONTHLY-WL - This file details for each major item in the study total monthly loss rates per WARF set including LOC, depot and inter-theater logistic losses. Figure 3.19.7 presents the file layout and example of the data for this file.

**19.1.2.B OUTPUT DATA AND DATA FILES:** The only output from this utility is the FINAL/REPORT. This report is a formatted, printed summary of daily loss rates of the Materiel Postprocessor (MPP) and the ELCON. The report is divided into two sections. The first section is a replay of the RATES-XX/SEC-1 file which documents the data that was provided to ELCON from the CONTROL-XX file. It should be noted again that in order for the FINAL/REPORT to run this RATES-XX/SEC-1 file must exist. In other words when ELCON was run, the Print Indicator field, cols. 31-35, in the Run Parameters Record of the CONTROL-XX file must not be set to "O".

The second portion of this report will provide for each major item of equipment in the study a period by period summary of its daily loss rates and quantity authorization levels. This portion of the report assumes the first eight WARF sets used by the ELCON as specified in the CONTROL-XX file were:

<u>WARF SET</u>	<u>PERIOD</u>
1	1st 15 Days
2	2nd 15 Days
3	1st 30 Days
4	2nd 30 Days
5	3rd 30 Days
6	4th 30 Days
7	5th 30 Days
8	6th 30 Days

WARF sets beyond 8 will be ignored. If less than 8 WARF sets are specified the remaining periods will be zero.

For each item of equipment in this section of the report three lines of information are used. The first line details daily loss rates for the item without LOC, depot and inter-theater logistic losses; the second line details daily loss rates for each period including shipping losses; the final line denotes the average per period quantity authorized for this item.

It should be emphasized that loss rates expressed in this report are daily loss rates rather than monthly loss calculated in the ELCON. To get these daily loss rates the monthly loss rates are simply divided by 30.

Figure III.9.8 presents an example of the FINAL/REPORT output.

### 19.1.2.C VARIABLE DICTIONARY:

The following section identifies and defines the various variables used in this utility.

<u>NAME</u>	<u>DEFINITION</u>
COPY	This variable is used as an array by the utility to copy and format data from the RATES-XX/SEC-1 input file directly to the main output file FINAL-REPORT-XX.
LIN	This 6 character alphanumeric variable will contain the Line Code of the current item under examination. This line code identifies the item within the system. This value is read from RATES-XX/MONTHLY-WOL input file and written directly to the FINAL/REPORT output file.
NOMEN	This 6 character alphanumeric variable is used as an array in the utility of 5 occurrences. The contents of the array will contain the 30 character nomenclature or description of the item of equipment being analyzed. This data will be read from RATES-XX/-MONTH-LY-WOL input file and written directly to the FINAL/REPORT-XX output file.
MORATE	This is an array with 8 occurrences used by the utility to hold the monthly attrition rates of the item of equipment being currently analyzed. This data is read from the RATES-XX/MONTHLY-WOL.
DAYRAT	This is an array of 8 occurrences used by the utility to store the results of the calculations to translate the monthly loss rates. This is accomplished by dividing each MORATE value by 30.
DUMMY	An array used to read past data from the input file, ITMID/FINAL, that is not required by the utility.
IQUAN	An integer array used by the utility stores the authorized quantities of the current item under analysis. This data is read from the ITMID/FINAL input file.
QUAN	This is an array used by the utility to store the average authorized quantities per time period of the study. In general these values are calculated by dividing the sum of this consecutive IQUAN values by 2.

II,LLL,KI,  
KJ,LI,KM,LN,  
IP,IQ,IR,IT

These various integers are used as subscripts in the utility.

**19.2 OPERATING ENVIRONMENT:** This program is implemented on the EXECUTIVE-8 operating system.

**19.2.1 SUPPORT SOFTWARE:** This report utility is written in FORTRAN IV and requires the FORTRAN IV compiler in addition to the UNIVAC 1100/82 system facilities.

**19.2.2 I/O DEVICES:** This report utility uses as input several files which reside on disk. The utility will produce a report first presumably on disk and then routed to a line printer for final output. Refer to Volume I for the runstream.

**19.3 MAINTENANCE PROCEDURES:** This program is maintained by the MPP analyst.

**19.3.1 PROGRAMMING CONVENTIONS:** The utility is written in FORTRAN IV and adheres to FORTRAN programming conventions. While this is a report program, no precautions have been taken to accomodate page breaks or page numbering.

**19.3.2 INTERNAL ERROR ROUTINES:** This report utility has no explicit error handling facilities. As a result only system detected errors will be identified and documented. Reference to system documentation will be required to correct errors.

# FINAL/REPORT STRUCTURE

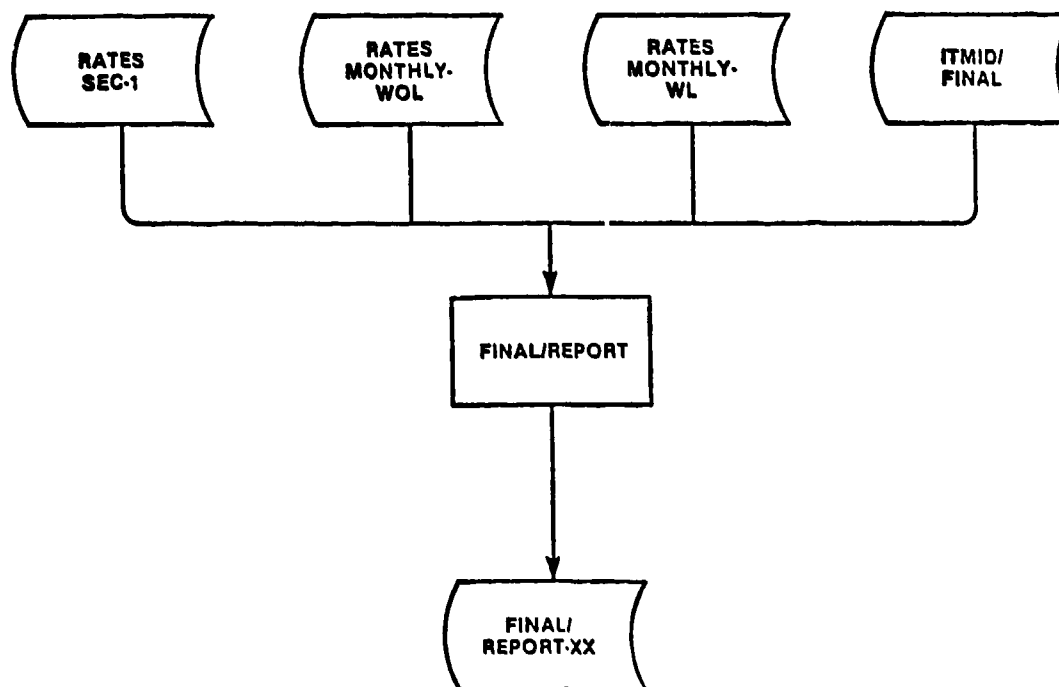
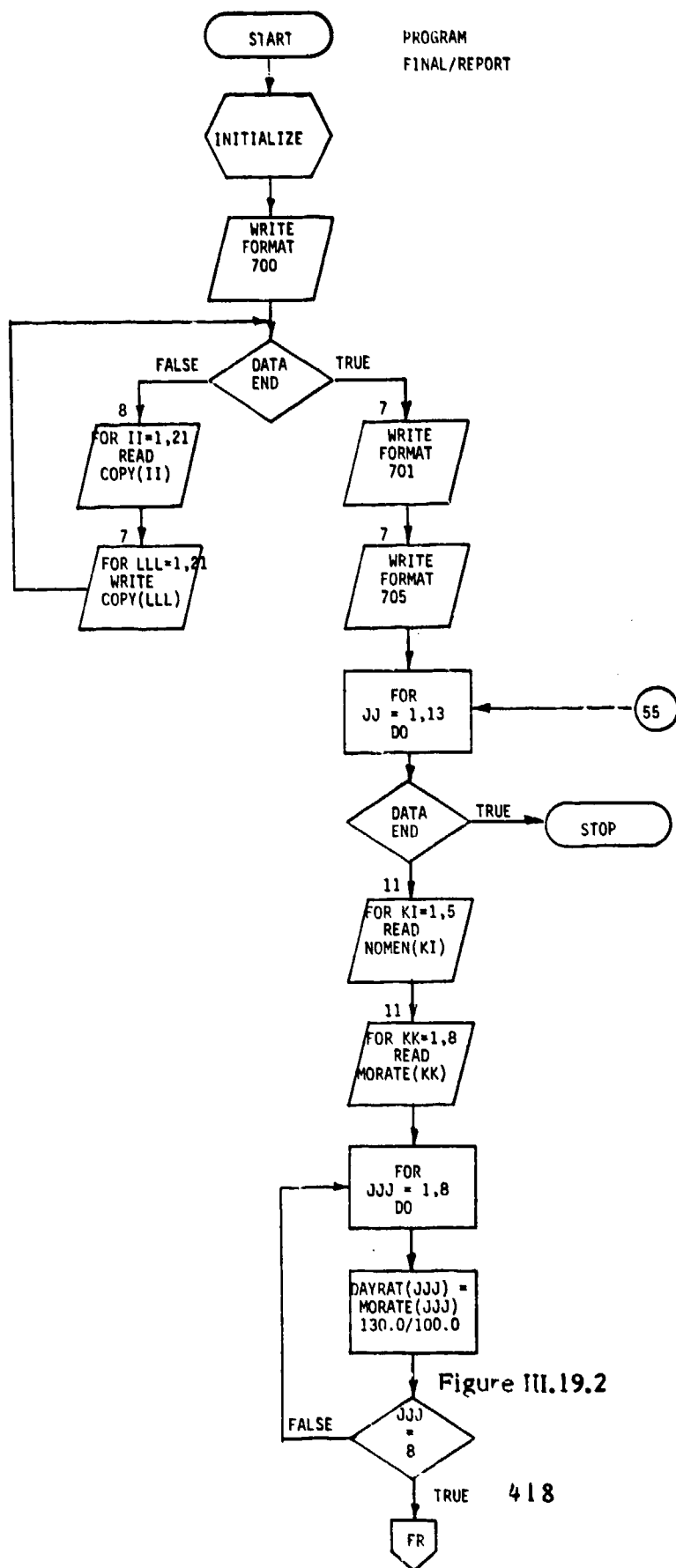


Figure III.19.1



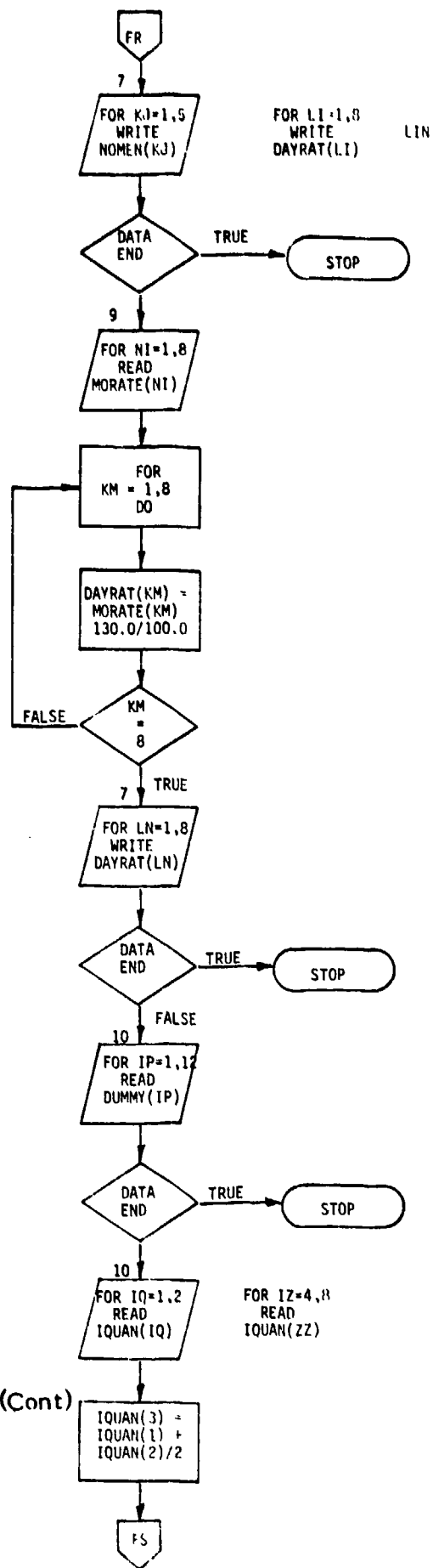


Figure III.19.2 (Cont)

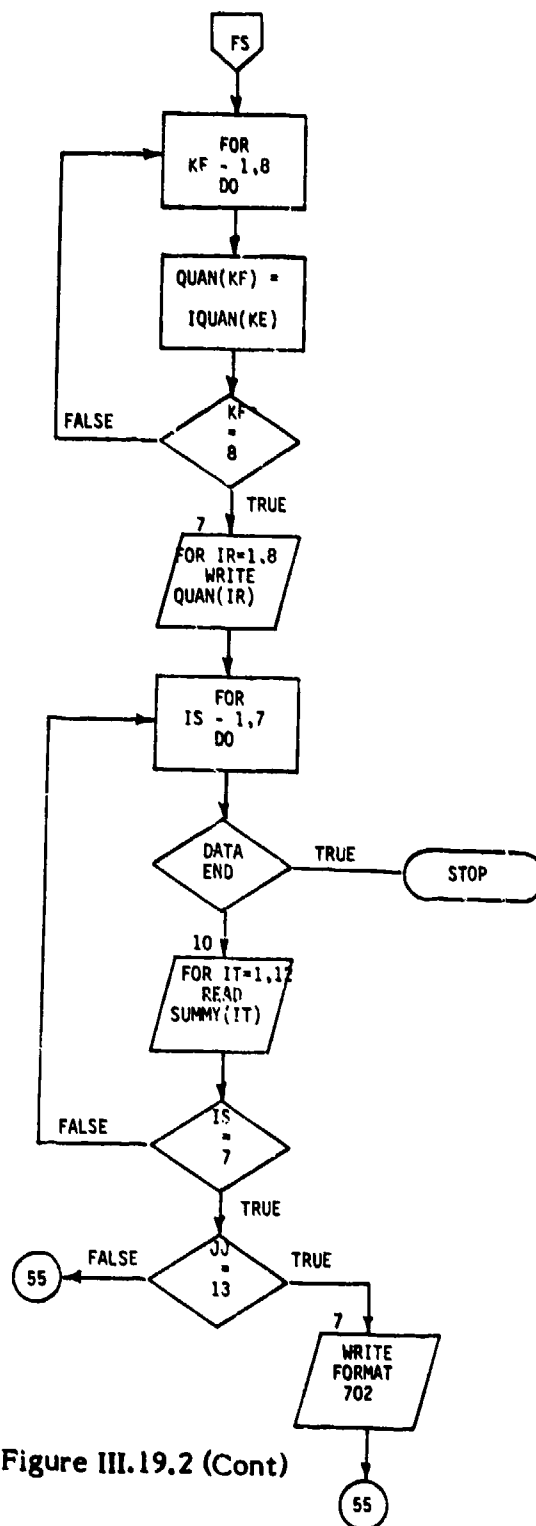


Figure III.19.2 (Cont)

UNCLASSIFIED\*\*\*FILE NAME:82XQT ELEMENT NAME:FINAL/REPORT\*\*\*UNCLASSIFIED

```

1:C*****
2:C*****
3:C*****THIS PROGRAM IS WRITTEN IN FORTRAN V.
4:C*****
5:C*****
6:C*****PURPOSE OF PROGRAM:
7:C*****
8:C*****THIS UTILITY IS DESIGNED TO TAKE THE OUTPUT OF FLCON
9:C*****AND PRODUCE A FINAL REPORT OF WARF DAILY ATRITION RATES.
10:C*****EACH LIN CODE WILL HAVE THREE LINES OF DATA. THE FIRST LINE
11:C*****WILL HAVE ATRITION RATES WITHOUT SHIPPING LOSSES AND THE
12:C*****SECOND WITH SHIPPING LOSSES INCLUDED AND THE LAST LINE WILL
13:C*****HAVE THE AVFRAGE AUTHORIZED QUANTITY OF THE LIN CODE IN
14:C*****THEATER FOR EACH TIME PERIOD.
15:C*****
16:C*****
17:C*****VARIABLE DICTIONARY
18:C*****VARIABLE NAME      DEFINITION
19:C*****NO                NUMBER OF CEM RUN
20:C*****COPY(I)           USED TO COPY LINE OF UNEFFECTED DATA
21:C*****LIN               LIN CODE
22:C*****NOMEN(IJ)         NOMENCLATURE OF EACH LIN CODE
23:C*****MORATE(K)         MONTHLY ATRITION RATE
24:C*****DAYRAT(L)         DAILY ATRITION RATE
25:C*****QUAN(I)           AVERAGE AUTHORIZED TOE QUANTITY BY PERIOD
26:C*****DUMMY(I)         DUMMY READ VARIABLE
27:C*****
28:C*****
29:      DIMENSION COPY(21),DAYRAT(8),DUMMY(12),QUAN(6),IQUAN(8)
30:      REAL LIN,NOMEN(5),MORATE(8)
31:      WRITE(7,700)
32:700    FORMAT(15X,'          DAILY ATRITION RATES')
33:99     READ(8,801,END=98)(COPY(I),I=1,21)
34:801    FORMAT(21A6)
35:      WRITE(7,801)(COPY(LLL),LLL=1,21)
36:      GO TO 99
37:98     WRITE(7,701)
38:701    FORMAT(9(1),35X,'DAILY ATRITION RATES WITHOUT & WITH LOG LOSSES')
39:      WRITE(7,705)
40:705    FORMAT(1X,'LIN',4X,'NOMENCLATURE',25X,'1ST15',5X,'2ND15',
41:      15X,'1ST30',5X,'2ND30',5X,'3RD30',5X,'4TH30',5X,'5TH30',5X,
42:      2'6TH30'////)
43:      GO TO 55
44:50     WRITE(7,702)
45:702    FORMAT(1X,'LIN',4X,'NOMENCLATURE',25X,'1ST15',5X,'2ND15',
46:      15X,'1ST30',5X,'2ND30',5X,'3RD30',5X,'4TH30',5X,'5TH30',5X,
47:      2'6TH30'////)
48:55     DO 20 JJ=1,13
49:      READ(11,1101,END=97)LIN,(NOMEN(KI),KI=1,5),(MORATE(KK),KK=1,8)
50:1101    FORMAT(1X,A6,1X,5A6,6F6.2,/)
51:      DO 30 JJJ=1,8
52:      DAYRAT(IJJ)=(MORATE(IJJ)/30.0)/100.0
53:30     CONTINUE
54:      WRITE(7,703)LIN,(NOMEN(KJ),KJ=1,5),(DAYRAT(LI),LI=1,8)
55:703    FORMAT(1X,A6,1X,5A6,1X,'W70L',F7.5,7(F10.5))
56:      READ(9,900,FND=97)(MORATE(NI),NI=1,8)
57:900    FORMAT(38X,PF6.2,/)

```

Figure III.19.3

UNCLASSIFIED\*\*\*FILE NAME:82XQT ELEMENT NAME:FINAL/REPORT\*\*\*UNCLASSIFIED

```
58:      DO 40 KM=1,8
59:      DAYRAT(KM)=(MORATE(KM)/30.0)/100.0
60:40     CONTINUE
61:      WRITE(7,704)(DAYRAT(LN),LN=1,8)
62:704     FORMAT(39X,'M L',F7.5,7(F10.5))
63:      READ(10,1000,END=97)(DUMMY(IP),IP=1,12)
64:1000    FORMAT(12A6)
65:      READ(10,1001,END=97)(IQUAN(IQ),IQ=1,2),(IQUAN(IZ),IZ=4,8)
66:1001    FORMAT(2I7,5I7)
67:      IQUAN(3)=(IQUAN(1)+IQUAN(2))/2
68:      DO 80 KF=1,8
69:      QUAN(KF)=IQUAN(KF)
70:80     CONTINUE
71:      WRITE(7,715)(QUAN(IR),IR=1,8)
72:715     FORMAT(19X,'AVG AUTH THEATER QTY',1X,8(F9.1,1X)/)
73:      DO 60 IS=1,7
74:      READ(10,1000,END=97)(DUMMY(IT),IT=1,12)
75:60     CONTINUE
76:20     CONTINUE
77:      GO TO 50
78:97     STOP
79:      END
```

Figure III.19.3 (Cont)

UNCLASSIFIED\*\*\*EXAMPLE OF THE OUTPUT OF UTILITY ITMID/REC-A\*\*\*UNCLASSIFIED

```

1: A03198 AK VEH M218 GM EQ P1A      0 522 30 0 1 2 1
2:   54      64      44      58      78      88      88
3: .00 .00 .00 .80 .20
4: .00 .00 .00 .80 .20
5: .00 .00 .00 .80 .20
6: .00 .00 .00 .80 .20
7: .00 .00 .00 .80 .20
8: .00 .00 .00 .80 .20
9: .00 .00 .00 .75 .25
10: A14752 ADAP TEST CAMERA LM178      01636 30 0 1 2 2
11:   10      16      16      17      17      17      17
12: .00 .00 .20 .80 .00
13: .00 .00 .20 .80 .00
14: .00 .00 .20 .80 .00
15: .00 .00 .15 .85 .00
16: .00 .00 .15 .85 .00
17: .00 .00 .35 .65 .00
18: .00 .00 .35 .65 .00
19: A22496 AIMING CIRCLE M2 W/E      01636 30 0 1 2 3
20:   6615   6699   8820   8823   8823   8823   8823
21: .25 .25 .50 .00 .00
22: .25 .25 .50 .00 .00
23: .25 .25 .50 .00 .00
24: .25 .25 .50 .00 .00
25: .25 .25 .50 .00 .00
26: .25 .25 .50 .00 .00
27: .25 .25 .50 .00 .00
28: A23770 AIR COND FL/WNDW 60008      01833 30 0 1 2 4
29:   0      0      0      0      0      0      0
30: .00 .00 .00 .00 .00
31: .00 .00 .00 .00 .00
32: .00 .00 .00 .00 .00
33: .00 .00 .00 .00 .00
34: .00 .00 .00 .00 .00
35: .00 .00 .00 .00 .00
36: .00 .00 .00 .00 .00
37: A23828 AIR COND F/WA 9000 RTU      01833 30 0 1 2 5
38:   889   993   998   998   998   998   998
39: .00 .00 .25 .25 .50
40: .00 .00 .25 .25 .50
41: .00 .00 .25 .25 .50
42: .00 .00 .25 .25 .50
43: .00 .00 .25 .25 .50
44: .00 .00 .25 .25 .50
45: .00 .00 .25 .25 .50
46: A24044 AIR COND 18000 BTU      01833 30 0 1 2 6
47:   53   53   53   55   55   55   55
48: .00 .00 .00 .50 .50
49: .00 .00 .00 .50 .50
50: .00 .00 .00 .50 .50
51: .00 .00 .00 .50 .50
52: .00 .00 .00 .30 .70
53: .00 .00 .00 .30 .70
54: .00 .00 .00 .30 .70
55: A24318 AIR COND 18000 BTU      01833 30 0 1 2 7
56:   15   25   42   57   63   83   97
57: .00 .00 .25 .75 .00

```

Figure III.19.4

UNCLASSIFIED\*\*EXAMPLE OF RATES-XX/SEC-1 OUTPUT DATA FROM UTILITY ELCON/3

```

1: TEMPORARY CONTROL DATA FILE NAME P88112DEC
2:
3:
4:
5:
6: NPER MXITH # CEM CLASSES #WARF SETS
7: 7 6 9 10
8:
9:
10: DAYS PER PERIOD
11: 15 15 30 30 30 30
12:
13:
14: FIRST AND LAST PERIOD FOR EACH WARF SET
15: 1 1
16: 2 2
17: 1 2
18: 3 3
19: 4 4
20: 5 5
21: 6 6
22: 7 7
23: 1 4
24: 5 7
25:
26:
27: AIR LOSS RATE BY PERIOD
28: .050 .050 .050 .010 .010 .000 .000
29:
30:
31: SEA LOSS RATE BY PERIOD
32: .150 .150 .230 .100 .050 .020 .000
33:
34:
35: LOC LOSS RATE BY PERIOD
36: .150 .150 .100 .100 .050 .050 .050
37:
38:
39: AVERAGE LARGE UNITS BY PERIOD
40: 48.10 51.50 52.00 52.00 52.00 52.00 52.00
41:
42:
43: FRACTION OF FORCE BY POSTURE BY PERIOD
44: ATTACK DEFEND WITHDRAW INACTIVE
45: .0000 .1410 .0590 .8000
46: .2550 .1450 .0500 .6000
47: .2000 .1550 .0050 .6400
48: .1000 .0550 .1050 .8400
49: .0050 .0550 .1400 .8000
50: .3050 .0050 .0000 .6900
51: .5550 .0000 .0000 .4450
52:
53:
54: ARTY KILLS (K PER DAY) BY VULN CLASS BY POSTURE
55:
56: .000 .994 .211 1.666 .222 .111 .666 .090 .138 .899 .574
57: .224 .691 .264 .714 .999 .744 .359 .807 .644 .921 .100
58:
59: .100 .992 .600 2.744 .578 .000 .299 .000 .704 .000 2.192
60: 7.682 3.339 .668 1.750 1.199 .502 .078 .515 .597 .000 .080
61:
62: .200 2.507 1.254 .927 .494 .000 .633 .000 .283 .284 1.609
63: 6.753 3.355 .895 3.001 1.678 1.789 .465 .898 .566 .000 .090
64:
65: .050 .255 .064 .555 .888 .010 .088 .550 .823 .760 .064
66: 3.142 7.460 .533 .850 .973 .349 .077 .050 .778 .000 .080
67:
68:
69:
70: ARTY SCALING FACTORS BY PERIOD
71: .649 .764 .980 1.000 .950 .485 .196
72:
73:
74: CEM KILLS (K PER 30 DAYS) BY CLASS BY PERIOD
75:
76: 64.000 1.400 .000 .000 10.000 24.000 36.000 72.000 7.800
77:
78: 56.000 1.400 .000 2.000 20.000 34.000 52.000 60.000 4.200
79:
80: 50.000 .900 .000 2.000 15.000 24.000 37.000 28.000 2.300
81:
82: 27.000 .500 .000 .000 6.000 18.000 25.000 10.000 .600
83:
84: 19.000 .400 .000 .000 5.000 13.000 16.000 4.000 .500
85:
86: 10.000 .300 .000 4.000 4.000 14.000 33.000 3.000 .100
87:
88: 17.000 .100 .000 2.000 3.000 7.000 15.000 1.000 .100

```

Figure III.19.5

UNCLASSIFIED\*\*\*EXAMPLE OF RATES-XX/MONTHLY-WOL OUTPUT DATA FROM UTILITY ELCON/3

1:	AD3198 AK VEH M218 6M E8 P1A	2.43	3.21	2.85	2.48	3.14	3.11	2.28	3.03	2.85	2.80
2:											
3:	A14752 ADAP TEST CAMERA LH178	8.04	10.65	9.64	9.79	10.48	10.26	8.08	7.06	10.00	8.47
4:											
5:	A22496 AIMING CIRCLE M2 W/E	30.73	40.37	35.58	39.79	49.34	46.76	21.56	18.44	42.10	28.86
6:											
7:	A23770 AIR COND FL/WNDW 6000B	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
8:											
9:	A23828 AIR COND F/WA 9000 BTU	1.15	1.77	1.48	1.44	2.14	2.10	.95	.87	1.69	1.31
10:											
11:	A24044 AIR COND 18000 BTU	.60	.74	.67	.47	.83	.74	.33	.47	.66	.51
12:											

Figure III.19.6

UNCLASSIFIED\*\*\*EXAMPLE OF THE RATES-XX/MONTHLY-WL OUTPUT DATA FROM ELCON/3

1:	A03198 AK VEH M218 0M EQ P1A	2.89	8.52	5.94	3.27	11.82	6.57	3.15	3.31	7.18	4.21
2:											
3:	A14752 ADAP TEST CAMERA LM178	9.70	24.23	18.63	13.13	17.44	12.51	9.14	7.84	16.54	9.77
4:											
5:	A22496 AIMING CIRCLE M2 W/E	37.08	49.58	43.34	57.60	73.84	56.98	24.38	20.51	59.07	33.28
6:											
7:	A23770 AIR COND FL/WN0W 6000 B	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
8:											
9:	A23828 AIR COND F/WA 9000 BTU	1.36	5.25	3.41	1.98	3.13	2.49	1.06	.94	2.89	1.47
10:											
11:	A24044 AIR COND 18000 BTU	.71	.87	.79	.62	2.31	.00	.37	.50	1.21	.58
12:											

Figure III.19.7

# DAILY ATTRITION RATES

CONTROL: DATA FILE WART F-88 (12DEC84)

WART DATA & DEF. CLASSES WART SETS  
7 2 9 10

DAYS PER PERIOD  
15 15 30 30 30 30

FIRST AND LAST PERIOD FOR EACH WART SET

1 1  
2 2  
3 2  
3 3  
4 4  
5 5  
6 6  
7 7  
4 4  
5 7

AIR LOSS RATE BY PERIOD  
.050 .050 .050 .010 .010 .000 .000

SEA LOSS RATE BY PERIOD  
.150 .150 .150 .100 .100 .020 .000

LOC LOSS RATE BY PERIOD  
.150 .150 .100 .100 .050 .050 .050

AVERAGE LARGE UNITS BY PERIOD  
45.10 54.50 52.00 52.00 57.00 57.00 57.00

FRACTION OF FORCE BY POSTURE BY PERIOD

ATTACK	DEFEND	WITHDRAW	UNKNOWN
.0000	.1410	.0590	.8000
.2550	.1450	.0500	.8000
.2000	.1550	.0050	.8400
.1000	.0550	.1050	.8400
.0950	.0550	.1400	.8000
.3050	.0050	.0000	.8900
.5550	.0000	.0000	.4450

ARTY KILLS (% PER DAY) BY VEH. CLASS BY POSTURE

TANK		INFANTRY		ARTILLERY		ENGINEER		SUPPORT		TOTAL	
.000	.994	.211	1.886	.222	.111	.886	.090	.138	.899	.574	
.224	.694	.264	.714	.999	.244	.359	.807	.644	.921	.100	
.100	.992	.200	1.744	.978	.100	.792	.800	.704	.800	1.192	
1.000	.838	.268	1.750	1.192	.267	.878	.747	.15	.800	.080	

Figure III.19.8

.000	2.507	1.264	.927	.494	.000	.433	.000	.233	.284	1.407
6.753	3.535	.095	3.004	1.678	1.789	.465	.070	.566	.000	.070
.000	.255	.064	.555	.883	.010	.088	.550	.823	.740	.064
3.112	7.440	.533	.050	.973	.347	.077	.050	.778	.000	.080

ARET LOSING FACTORS BY PERIOD  
.547 .744 .930 1.000 .950 .485 .196

PER LOSS (% PER 30 DAYS) BY CLASS BY PERIOD

64,000	1,400	.000	.000	10,000	24,000	36,000	72,000	7,000
56,000	1,400	.000	2,000	26,000	34,000	52,000	60,000	4,200
50,000	.900	.000	2,000	15,000	24,000	37,000	28,000	2,300
27,000	.500	.000	.000	6,000	18,000	25,000	10,000	.600
19,000	.400	.000	.000	5,000	13,000	16,000	4,000	.500
20,000	.300	.000	4,000	4,000	14,000	33,000	3,000	.400
17,000	.100	.000	2,000	3,000	7,000	15,000	1,000	.100

LTN	DESCRIPTION	DATE ATTRITION RATES WITHOUT AND WITH LOSSES							
		15115	24015	15130	24030	30030	4TH30	5TH30	6TH30
A03100 HIL VEH N211 OR ED P1A	W/O L	.0080	.0011	.0010	.0080	.0010	.0010	.0008	.0010
	M L	.0010	.0028	.0020	.0011	.0039	.0022	.0011	.0011
	AVE QTY	59.0	54.0	49.0	51.0	64.0	83.0	88.0	53.0
A20720 AIR COND FLANDW 6000R	W/O L	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
	M L	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
	AVE QTY	.0	.0	.0	.0	.0	.0	.0	.0

Figure III.19.8 (Cont)

## Appendix A

### References

- A. COSAGE Data Requirements Document, 1st Ed, USACAA, Documentation CAA-D-80-8, dated 1 Jun 80.
- B. Wartime Requirements for Ammunition, Materiel and Personnel (WARRAMP) Five-year Master Plan, CAA Memorandum, 12 October 1979.
- C. Wartime Requirements for Ammunition, Materiel and Personnel (WARRAMP) Analytical Methodology, Technical Paper, CAA-TP-78-4, USACAA, dated August 1978.
- D. Equipment Loss Consolidator (ELCON) Methodology, Documentation, CAA-D-79-3, USACAA, dated August 1979.
- E. The Forewon Force-Planning System, Appendix C, User's Guide for Force Analysis Simulation of Theater Administrative and Logistic Support (FASTALS), Documentation, Research Analysis Corporation, RAC-R-86, dated May 1971.
- F. Army Automation, Documentation Standards for Development and Maintenance of computer Software Memorandum, USACAA, CAA Memorandum 18-2, dated 26 October 1979.
- G. Army Automation, Software Design and Development, HQDA, TB 18-103, dated January 1980.
- H. Army Automation, Technical Documentation, HQDA, TB 18-111 dated January 1979.
- I. Automated Data Systems Documentation Standards, Department of Defense, DOD Standard 7935.1-S, dated 13 September 1977.
- J. System for Estimating Materiel Wartime Attrition and Replacement Requirements (SYMWAR), Vols I and II, RAC-TP-404, Sep 70 and Jan 71, respectively (on file at DDC).
- K. Selected Analysis of 1972-1973 Equipment Attrition Factors, Vols I and II, ODCSLOG, II Apr 73.
- L. An Analysis of Methods for Determining Wartime Materiel Requirements, Phase I, Vols I and II, CAA, Jun 73.
- M. Wartime Active Replacements Factors (WARF), Phase II, System Design, CAA, Jan 74.
- N. Wartime Replacement Factors--FY 80 (WARF-80), CAA (CAA-SR-75-9), Jul 75.
- O. Wartime Replacement Factors (WARF) Systems Documentation, CAA, Apr 76.

- P. Wartime Replacement Factors--FY 78-82 (WARF 78-82), Vol II, Main Report, CAA (CAA-SR-78-13), Oct 78.
- Q. Wartime Requirements for Ammunition and Materiel, FY 81-85. (AMMO P-85/WARF-85), Vol. I, Main Report, CAA (CAA-SR-79-1), Feb 79.
- R. Wartime Requirements for Ammunition and Materiel, FY 82-86 (AMMO P-86/WARF-85), Vol I, Main Report, CAA.
- S. SIMSCRIPT II.5 Programming Language, Kiviat, et.al; edited by E.C. Russell, 1975, CACI, Inc., 12011 San Vicente Blvd, Los Angeles, CA 90049
- T. SIMSCRIPT II.5 Reference Handbook, 1976, CACI, Inc., 12011 San Vicente Blvd, Los Angeles, CA 90049
- U. SIMSCRIPT II.5 User's Manual - UNIVAC 1100 Series Computer Systems 1977, CACI, Inc., 12011 San Vicente Blvd, Los Angeles, CA 90049.
- V. Target Acquisition Model, USACAA, Documentation CAA-D-74-2, dated December 1974.
- W. Red Artillery Model, USACAA, Documentation CAA-D-74-2, dated December 1974.
- X. Sperry UNIVAC 1100 Series Executive System Programmer Reference Manual, Volumes 1 - 4, UP 4144.1 through UP 4144.4, 1975 Sperry Rand Corporation, with updates.
- Y. Force Analysis Simulation of Theater Administrative and Logistics Support Model (FASTALS), prepared for the U.S. Army Logistics Center by Computer Sciences Corporation, April 1980, Contract GS-045-22715.

## Appendix B

### Terms and Abbreviations

APP	Ammunition Post Processor - A related group of computer software programs that is a part of the WARRAMP methodology; used to compute the expected consumption of ammunition of selected calibres of a force in a conflict.
CEM	Concepts Evaluation Model - A low resolution theater combat model that simulates the combat between two opponent forces over a specific period of time producing force results.
COSAGE	Combat Sample Generator, a high resolution model that simulates tactical combat between a red and blue force; a production model that produces force on force results.
ELCON	Equipment Loss Consolidator.
ESD	Equivalent Stylized Day of (Wartime) combat between a postured blue and red force; used to provide an activity comparison between forces.
HMS	Heavy Materiel Supply Units (Companies).
ITMID	Item Identification File.
K-KILL	A catastrophic kill of the item (target) rendering it incapable of returning fire or movement and is non-repairable.
LA	Lethal area of indirect fire (area type) weapon systems.
LEA	Logistics Evaluation Agency.
LIN	Line Item Number (Code) - LINCODE.
LOC	Lines of Communications.
MIE	Major items of equipment.
M-KILL	A hit on an item (target) that renders it immobile, but repairable and capable of returning fire.
ODCSOPS	Office of the Deputy Chief of Staff (Army) for Operations.
PK	Probability of Kill.
RAM	Red Artillery Model.
RTD	Returned To Duty; personnel or repaired equipment.

SSPK	Single Shot Probability of Kill.
SRC	Standard Requirements Code.
TAM	Target Acquisition Model.
TOE	Table of Organization and Equipment.
TRCONS	Theater Rate Consolidation data file.
TRMAP	Theater Rate Mapping data file.
TRM	Theater Rates Model, used to simulate a theater conflict, generating stylized combat periods; used to compute ammunition consumption rates for several weapon - munition combinations.
WARF	Wartime Replacement Factors, also known as Wartime Active replacement factors. Rates of loss or specified periods or time increments for selected combat materiel items.
WIMP	WARF Intermediate Materiel Processor.

**Appendix C  
Distribution**

<u>Addressee</u>	<u>Copies</u>
Deputy Under Secretary of the Army (Operations Research) HQ DA, Room 2E261 Pentagon Building Washington, DC 20310	1
Deputy Chief of Staff for Operations and Plans (DAMO-ZD) Room 3A538 Pentagon Building Washington, DC 20310	2
Commander US Army Operational Test and Evaluation Agency 5600 Columbia Pike Falls Church, Virginia 22041	1
Commander US Army Training and Doctrine Command (TRADOC) Fort Monroe, Virginia 23651	1
Commander US Army Combined Arms Combat Developments Activity Building 574 Fort Leavenworth, Kansas 66027	1
Commander US Army Combat Developments Experimentation Command Fort Ord, California 93941	1
Commander HQ USAREUR and 7A ATTN: DCSRM APO NY 09403	2
Defense Documentation Center for Scientific and Technical Information Building 5 Cameron Station Alexandria, Virginia 22314	5

The Army Central Library (ASDIRS)  
Room 1A526  
Pentagon Building  
Washington, DC 20310

1

Local

Chief, AD, ATTN: Library Branch  
Director, FA  
Director, JF  
Director SM  
Director, MC  
Director, RQ

2  
2  
2  
2  
12  
15